偏微分方程式の解作用素の近似理論
―作用素学習とニューラル作用素―

静岡大学・工学部
谷口 晃一

Koichi Taniguchi
Faculty of Engineering,
Shizuoka University

概要

ニューラル作用素は, 関数空間の間の写像を学習する深層学習モデルであり, 偏微分方程式の解作用素を近似する手法として近年注目を集めている. 特に, 一度学習されたモデルは, 新たな入力に対して高速に解を出力できるため, 数値ソルバーの代理モデルとして期待されている.

本稿では, [15] で示された, ニューラル作用素による非線形放物型方程式の解作用素の定量的近似定理を解説する. 前半では, 作用素学習, ニューラル作用素, 万能近似定理, およびパラメータ複雑性の呪いについて概観する. 後半では, 非線形放物型方程式に対象を絞り, Duhamel 原理と Picard 反復に基づく解作用素の構造を用いることで, 一般の作用素学習で問題となるパラメータ複雑性の呪いを回避しうる定量的近似定理が得られることを説明する. とりわけ, ニューラル作用素の層構造と Picard 反復との対応が本質的な役割を果たすことを強調する. 本稿の視点は, 放物型方程式に限らず, Picard 反復により解を構成できる他の方程式群にも有用であることを示唆する.

1 はじめに

科学技術計算と機械学習の融合は, 近年 Scientific Machine Learning (SciML) や AI for Science と総称され, 大きな注目を集めている. この交差領域の研究は, 大きく二つの方向に分けられる. 第一は, 科学技術計算の課題を解決あるいは効率化するために機械学習を用いる方向であり, Physics-Informed Neural Networks (PINNs) [45] や Sparse Identification of Nonlinear Dynamics (SINDy) [5] などがその代表例である. 本稿の主題である作用素学

RIMS 共同研究（公開型）「関数空間論とその周辺」2025 年 12 月

習 [37, 34] もこの流れに属し, 偏微分方程式 (PDE) の解作用素をデータから学習することで, 数値ソルバーの代理モデルを構成することを目指す. 第二は, 微分方程式や力学系の知見を学習理論の構築やモデルの解析に取り込む方向であり, Neural ODE [9] やリザバー計算 [22, 38] がその代表例である. さらに, 確率的勾配降下法を確率微分方程式で近似する枠組み [40, 36], 積分表現理論やリッジレット解析によるニューラルネットワークの表現・近似・構造理解 [41, 48, 47] など, 深層学習に数学的理解を与える試みも活発に進められている. この分野全体の概観については [2, 14, 20, 23, 24, 42, 52, 55] を参照されたい.

本稿では, このうち作用素学習, とりわけ PDE の解作用素の近似理論に焦点を当てる. 作用素学習では, 初期条件, 境界条件, 係数関数などの入力データから対応する解を与える写像, すなわち解作用素そのものを近似対象とする. 一度学習が達成されれば, 新たな入力に対して従来の数値計算を繰り返すことなく, 高速に解を出力できるため, 数値ソルバーの代理モデルとして期待されている. これまで, DeepONet [37, 30], PCA-Net [3, 29], ニューラル作用素 [34, 35] をはじめとして, さまざまなアーキテクチャが提案されてきた. これらの手法は, 流体方程式, 拡散方程式, 気象・気候モデルなどに対して経験的に高い性能を示しており, PDEBench に代表されるベンチマークでも広く検証されている [50].

もっとも, 作用素学習の理論基盤はなお発展途上にある. 近似理論の観点からは, コンパクト集合上での連続作用素に対する万能近似性, 各モデルが近似可能な作用素クラスの特徴づけ, 所望の精度を達成するために必要なモデル複雑性の評価, さらに一般の作用素に対して現れるパラメータ複雑性の呪いをいかに回避するか, といった問題が本質的な課題である. また, 学習理論の観点からは, 汎化性能や学習の安定性の解析も重要である. 加えて, PDE の解作用素がもつ構造や性質を, どのようにモデル設計や学習に反映させるかという点も重要である.

本稿では, これらのうち, 特にニューラル作用素の近似理論に焦点を当てる. 前半では, 作用素学習の基本的な考え方と代表的なアーキテクチャを紹介し, 万能近似定理 (Universal Approximation Theorem) およびパラメータ複雑性の呪い (Curse of Parametric Complexity) という二つの基本問題を概説する. 後半では, 非線形放物型 PDE の解作用素に対象を絞り, [15] の主結果を紹介する. 特に, Duhamel 原理と Picard 反復に基づく解の反復構造をニューラル作用素の層構造と対応づけることにより, 深さやニューロン数の指数的増大を回避しうる定量的近似定理が得られることを説明する. 本稿では, 学習アルゴリズムや実装, 数値検証の詳細には立ち入らず, PDE の構造を活用した近似理論の考え方に重点を置く.

2　作用素学習とニューラル作用素

本節では, 作用素学習の基本的事項について述べる. 作用素学習のより詳しい解説については, 例えば [28, 32] を参照されたい.

2.1　作用素学習

通常の深層学習では, ニューラルネットワークを用いて, 有限次元空間 $\mathbb{R}^{d_{\text{in}}}$ から $\mathbb{R}^{d_{\text{out}}}$ への写像をデータから近似する. これに対し, 作用素学習では, 無限次元関数空間 X から Y への写像, すなわち作用素

$$\mathcal{G} : X \to Y$$

を学習する. 特に近年は, PDE の解作用素をデータから学習し, 新たな入力に対する解を高速に出力するモデル構成法として, 作用素学習が大きな注目を集めている. このような発想は少なくとも Chen-Chen [7, 8] に遡ることができ, 近年では DeepONet, PCA-Net, ニューラル作用素などの枠組みのもとで再び活発に研究されている.

もっとも, 実際の計算で利用できるのは, 入力関数や出力関数そのものではなく, 有限個の格子点やセンサー点での値である. したがって, 作用素学習は本質的に「無限次元空間上の写像の学習」と「有限次元データからの推定」との間にあるギャップを扱う問題である. このとき重要なのは, 学習の対象が個々の離散化そのものではなく, その背後にある関数空間上の写像であるという点である. 換言すれば, メッシュや解像度に依存しない形で近似理論を定式化することが本質的に重要となる. この離散化不変性 (あるいは解像度頑健性) こそが, 作用素学習を単なる高次元回帰と区別する本質的特徴である [27, 28].

本節では, 作用素学習の代表的な枠組みである DeepONet, PCA-Net, ニューラル作用素について概説する. 簡単のため, 本稿では入出力関数はスカラー値とする. なお, ベクトル値関数の場合も各成分ごとに同様に定式化できる.

2.2　DeepONet と PCA-Net

作用素学習の代表的手法の一つが DeepONet である [37, 30]. DeepONet の基本的な考え方は, 作用素 $\mathcal{G}$ の出力 $\mathcal{G}(u)(x)$ を

$$\mathcal{G}(u)(x) \approx \sum_{j=1}^{J} c_j\big(u(x_1), \dots, u(x_m)\big)\, b_j(x)$$

の形で近似することにある. ここで, $x_1, \dots, x_m$ は有限個のセンサー点とし, c_j は入力関数 u に依存する係数, b_j は基底的関数である.

実装上, DeepONet は二つのネットワークから構成される. 一つは, センサー点 $x_1, \dots, x_m$ で観測された

$$\big(u(x_1), \cdots, u(x_m)\big)$$

を入力として係数

$$(c_1, \cdots, c_J)$$

を出力する branch network であり, もう一つは, 出力点 x を入力として

$$\big(b_1(x), \cdots, b_J(x)\big)$$

を出力する trunk network である. 最終的に, 両者の出力を組み合わせることで, 上式のような分離表現により作用素 $\mathcal{G}$ を近似する. このように DeepONet は入力関数 u に依存する情報と, 出力点 x に依存する情報とを分けて表現しており, 作用素のデータ駆動的な低ランク近似, あるいは非線形変数分離とみなすことができる.

もう一つの代表的枠組みが PCA-Net である [3, 29]. その基本的な考え方は, 出力関数の分布に対して主成分分析 (PCA) を施し, 主要なモードに沿って作用素を近似することにある. 具体的には,

$$\mathcal{G}(u)(x) \approx \overline{g}(x) + \sum_{j=1}^{J} c_j(u)\, \phi_j(x)$$

の形を考える. ここで, $\overline{g}$ は出力関数の平均, $\{\phi_j\}_{j=1}^{J}$ は平均を差し引いた出力分布に対する主成分基底である. PCA-Net では, 係数関数 c_j を有限次元ニューラルネットワークによって学習する. したがって PCA-Net は, 出力側の低次元構造に基づくモデル縮約を, 作用素学習の形で実現していると理解できる.

DeepONet と PCA-Net はいずれも有力な方法であるが, PDE の解作用素は一般に非局所性をもつ. ある一点での解の値は, 初期値や係数関数の局所情報だけでなく, 領域全体や過去の時刻における大域的情報に依存することが多い. このような構造をより直接的に反映するモデルとして, 自然に現れるのが, 積分作用素を中間層に組み込んだニューラル作用素である.

2.3 ニューラル作用素

ニューラル作用素は, 有限次元ベクトルではなく, 関数そのものを各層で更新することを基本思想とする [34, 35, 27]. 典型的には, 各中間層は

$$v^{(\ell+1)}(x) = \sigma\Big(W^{(\ell)} v^{(\ell)}(x) + \big(K^{(\ell)} v^{(\ell)}\big)(x) + b^{(\ell)}(x)\Big), \qquad \ell = 0, 1, \dots, L-1$$

で与えられる. ここで, $W^{(\ell)}$ は局所的な線形変換, $b^{(\ell)}$ はバイアス項, $K^{(\ell)}$ は非局所的な積分作用素, σ は活性化関数である. また, 簡単のため入出力関数はスカラー値としているが, 中間層 $v^{(\ell)}$ は一般に $\mathbb{R}^{d_\ell}$-値関数である.

非局所項 $K^{(\ell)}$ は典型的には

$$\big(K^{(\ell)}v\big)(x) = \int_D \kappa^{(\ell)}(x,y)\, v(y)\, dy$$

の形で与えられ, 積分核 $\kappa^{(\ell)}(x,y)$ が点 y の情報が点 x にどのように寄与するかを記述する. 時間発展方程式を視野に入れる場合は,

$$\big(K^{(\ell)}v\big)(t,x) = \int_0^T \int_D \kappa^{(\ell)}(t,\tau,x,y)\, v(\tau,y)\, dy\, d\tau$$

のような時空間上の積分作用素を考えるのが自然である.

理論面でも実装面でも重要なのは, この積分作用素を有限ランク近似することである. たとえば，適当な関数族 $\{\varphi_n\}_{n\in\Lambda}$，$\{\psi_m\}_{m\in\Lambda}$ を用いて積分核を

$$\kappa^{(\ell)}(x,y) = \sum_{m,n\in\Lambda} C^{(\ell)}_{mn}\, \varphi_n(x)\psi_m(y)$$

と近似すれば,

$$\big(K^{(\ell)}_N v\big)(x) := \sum_{m,n\in\Lambda_N} C^{(\ell)}_{mn}\, \langle \psi_m, v\rangle\, \varphi_n(x)$$

という有限ランク作用素が得られる. ここで, $\Lambda_N \subset \Lambda$ は要素数 N の有限集合であり, $\langle \psi_m, v\rangle$ は内積あるいは双対積を表す. このように, ニューラル作用素は, 有限ランク積分作用素を各層に組み込んだ深層モデルとして理解することができる. したがって, その複雑性を議論する際には, 深さやニューロン数だけでなく, 打ち切りランク N も考慮する必要がある.

この構造は, 後に述べる Duhamel 原理や Green 関数の基底展開と極めて相性がよい. 実際, 多くの PDE は積分方程式あるいは積分作用素の形で記述できるため, 対応する積分核を適切な基底で近似することにより, ニューラル作用素の各層を PDE の反復解法と対応づけることが可能となる. 本稿でニューラル作用素を中心に扱う理由は, まさにこの点にある.

2.4 代表的なニューラル作用素：FNO と WNO

ニューラル作用素の具体的アーキテクチャは, 主に非局所項 $K^{(\ell)}$ の実装方法によって区別される. その代表例が Fourier Neural Operator (FNO) [35] と Wavelet Neural Operator (WNO) [18] である

FNO では, 非局所項を Fourier 空間での乗算として実装する. 具体的には,

$$K^{(\ell)}v = \mathcal{F}^{-1}\big(R^{(\ell)}\,\mathcal{F}v\big)$$

と定める. ここで $\mathcal{F}$ は Fourier 変換, $\mathcal{F}^{-1}$ は Fourier 逆変換, $R^{(\ell)}$ は学習可能な線形変換である. 通常は低周波モードに制限した形で $R^{(\ell)}$ を学習し, 高周波成分は切り捨てるか, あるいは別の局所項で補う. この構造により, 非局所的相互作用を効率的に扱うことができ, FFT を用いた高速計算も可能となる [35, 26, 27].

一方 WNO は, wavelet 基底を用いて作用素を表現する [18, 51]. Wavelet は位置とスケールの両方に局在した基底であるため, 多重スケール構造や局所的変化を扱ううえで有利であると期待される. Fourier 基底が全域的な振動モードを表現するのに対し, wavelet 基底は空間局所性を保ったままスケール分解を行うため, 境界の影響や局所的特異構造が重要となる問題に適している可能性がある.

さらに, 作用素学習の枠組みはユークリッド空間に限られない. たとえば, 球面上の FNO [4] やリーマン多様体上のニューラル作用素 [10] が提案されており, 基底展開や積分核の選択を通して, 幾何学的構造を反映した作用素学習へと自然に拡張できることが示されている.

2.5 本稿の立場

ここまで見たように, 作用素学習には DeepONet, PCA-Net, ニューラル作用素など, いくつかの代表的枠組みが存在する. しかし, 本稿の関心は「どのモデルが数値実験で最も高精度か」を比較することにはない. 本稿の関心はむしろ「PDE の解作用素の良い近似モデルをどのように設計すべきか」, 「そのために PDE のどのような構造が本質的なのか」を理論的に理解することにある. この観点から見ると, 単なる万能近似性だけでは不十分であり, 重要なのは, 所望の精度 $\varepsilon > 0$ を達成するために必要な深さ, ニューロン数, ランクなどの複雑性がどのように増大するかを定量的に把握することである. 実際, 一般の連続作用素に対しては, 後述するようにパラメータ複雑性の呪いが現れることが知られている. したがって, この困難を回避するためには, 近似対象を PDE の解作用素のような特別な構造をもつ作用素へと制限し, その構造をモデル側に取り込む必要がある. ニューラル作用素は, 積分作用素, Green 関数, 半群性, 反復解法といった PDE 側の構造を直接反映できるという点で, この目的に適した有力な枠組みである. 本稿では, まさにこの観点からニューラル作用素の近似理論を扱う.

3　万能近似性とパラメータ複雑性

本節では, ニューラル作用素の近似理論の出発点となる万能近似性とパラメータ複雑性の呪いについて簡単に紹介する. 重要なのは, 万能近似性が「近似できる」ことを保証しても, 「効率よく近似できる」ことまでは保証しないという点である. 近似に必要なパラメータ数について, 有限次元では Yarotsky 型の多項式的な下界が現れ, 無限次元では Lanthaler-Stuart 型の指数関数的な下界が現れる. このことが, 第 4 節で PDE の特別な構造を用いる動機となる.

3.1　有限次元の場合：万能近似性と次元の呪い

まず, 有限次元空間における深層ニューラルネットワークを考える. 入力空間を $\mathbb{R}^d$, 出力空間を $\mathbb{R}$ とし, 活性化関数 $\sigma : \mathbb{R} \to \mathbb{R}$ を固定する. 層の深さ $L \in \mathbb{N}$ に対し, $d_0 = d$, $d_{L+1} = 1$ として, 各層の幅を $d_1, \dots, d_L$ とすると, 深層ニューラルネットワーク $h : \mathbb{R}^d \to \mathbb{R}$ は

$$
\begin{cases}
z^{(0)} = x, \\
z^{(\ell+1)} = \sigma\big(W^{(\ell)} z^{(\ell)} + b^{(\ell)}\big), \qquad \ell = 0, 1, \dots, L-1, \\
h(x) = W^{(L)} z^{(L)} + b^{(L)}
\end{cases}
$$

の形で与えられる. ここで, $W^{(\ell)} \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$, $b^{(\ell)} \in \mathbb{R}^{d_{\ell+1}}$ であり, 活性化関数 σ は各成分ごとに作用するものとする.

$$
\mathcal{A} = (\sigma, L, d_0, \dots, d_{L+1})
$$

をネットワークのアーキテクチャと呼び, アーキテクチャ $\mathcal{A}$ に対応するネットワーク全体を $\mathcal{H}(A)$ で表す. すなわち, $\mathcal{H}(\mathcal{A})$ は重みとバイアスを自由に動かして得られるニューラルネットワーク全体の集合である. さらに, 許容されるすべてのアーキテクチャに対して

$$
\mathcal{H} := \bigcup_{\mathcal{A}} \mathcal{H}(\mathcal{A})
$$

とおく.

機械学習の近似理論において, まず問われるのは, この仮説集合 $\mathcal{H}$ がどの程度豊かな表現能力をもつかという点である. その最も基本的な結果が万能近似定理である.

定理 3.1 (Cybenko [12], Funahashi [16])**.** $K \subset \mathbb{R}^d$ をコンパクト集合とし, σ を連続なシグモイド関数とする. このとき, 任意の連続関数 $f \in C(K)$ と任意の $\varepsilon > 0$ に対して, ある浅いニューラルネットワーク $h \in \mathcal{H}$ が存在して

$$
\sup_{x \in K} |f(x) - h(x)| \le \varepsilon
$$

が成り立つ. すなわち, $\{h|_K : h \in \mathcal{H}\}$ は $C(K)$ において稠密である.

定理 3.1 は隠れ 1 層ネットワークに対する古典的結果である. したがって, 浅いネットワークを特別な場合として含む深いネットワークも同様の万能近似性をもつ. また, シグモイド関数に限らず, 有界非定数活性化関数, 非多項式活性化関数, ReLU 活性化関数のような非有界関数など, より広いクラスの活性化関数に対しても万能近似定理が成立することが知られている [13, 19, 25, 33, 44].

しかし, 近似可能であることと, それが効率的に実現できることは全く別問題である. 実際, 入力次元 d が大きくなると, 所望の精度を達成するために必要なパラメータ数が急増する, いわゆる「次元の呪い」が現れることがある. このことを示す代表的結果の一つが Yarotsky [53] による ReLU ネットワークの複雑性下界である.

定理 3.2 (Yarotsky [53])**.** $d, r \in \mathbb{N}$ とし,

$$\mathcal{F}_{d,r} := \{f \in W^{r,\infty}([0,1]^d) : \|f\|_{W^{r,\infty}} \leq 1\}$$

とする. 各 $\varepsilon \in (0,1)$ に対し, ある ReLU ネットワークのアーキテクチャ $\mathcal{A}_\varepsilon$ が存在して, 任意の $f \in \mathcal{F}_{d,r}$ に対してある $h_f \in \mathcal{H}(\mathcal{A}_\varepsilon)$ が存在し,

$$\sup_{x \in [0,1]^d} |f(x) - h_f(x)| \leq \varepsilon$$

が成り立つとする. このとき, $\mathcal{A}_\varepsilon$ に対して次が成り立つ.

(i) ある定数 $c > 0$ が存在して

$$W(\mathcal{A}_\varepsilon) \geq c\, \varepsilon^{-d/(2r)}$$

が成り立つ.

(ii) さらに, ある $p \geq 0$ と $C > 0$ が存在して

$$L(\mathcal{A}_\varepsilon) \leq C(\log(1/\varepsilon))^p$$

が成り立つならば, ある定数 $c' > 0$ が存在して

$$W(\mathcal{A}_\varepsilon) \geq c'\, \varepsilon^{-d/r} (\log(1/\varepsilon))^{-2p-1}$$

が成り立つ.

ただし, $W(\mathcal{A}_\varepsilon)$ と $L(\mathcal{A}_\varepsilon)$ は, それぞれ $\mathcal{A}_\varepsilon$ の非零重みの個数と層数を表す.

定理 3.2 は, 滑らかな関数クラスに対してさえ, 近似精度 ε に応じて必要なネットワークサイズが入力次元 d に依存して多項式的に増大することを示している. したがって, 万能近似性だけでは不十分であり, 「どのような関数クラスに対して, どのようなレートで近似できるか」を問う定量的理論が重要となり, 盛んに研究されている. 例えば, Barron 型関数クラスに対する近似 [1], 深い ReLU ネットワークによる近似率 [53, 46], Sobolev 空間および Besov 空間に対する近似誤差評価 [17, 49] などが知られている.

3.2 無限次元の場合：万能近似性

次に, 関数空間の間の写像に対する万能近似性を述べる. X, Y を Banach 空間, $K \subset X$ をコンパクト集合とする. 連続作用素

$$\mathcal{G} : K \to Y$$

に対して, ニューラル作用素の万能近似定理は次のように述べられる.

定理 3.3 (簡略形)**.** *X, Y を Banach 空間, $K \subset X$ をコンパクト集合とし, $\mathcal{G} : K \to Y$ を連続作用素とする. このとき, 任意の $\varepsilon > 0$ に対して, あるニューラル作用素 Γ_ε が存在して*

$$\sup_{u \in K} \|\mathcal{G}(u) - \Gamma_\varepsilon(u)\|_Y < \varepsilon$$

が成り立つ.

注意 3.4. 上の定理は, 本稿の目的に合わせて簡略化した形で述べている. 既存文献では, 具体的な関数空間 ($C^m, L^p, W^{m,p}$ など) と, 積分核や基底展開のクラスを仮定した形でより精密に述べられている. また, 同様の万能近似定理は DeepONet や PCA-Net に対しても知られている. 厳密な主張は [26, 27, 28, 29, 30, 32, 37] などを参照されたい.

定理 3.3 の証明の基本的アイデアは, 無限次元の作用素近似を有限次元の問題へ帰着することにある.

補題 3.5 ([27])**.** *X, Y を Banach 空間, $K \subset X$ をコンパクト集合とし, $\mathcal{G} : K \to Y$ を連続作用素とする. このとき, 任意の $\varepsilon > 0$ に対して, ある整数 $J, J' \in \mathbb{N}$, 連続線形写像*

$$F : X \to \mathbb{R}^J, \qquad E : \mathbb{R}^{J'} \to Y$$

および連続写像

$$\psi : \mathbb{R}^J \to \mathbb{R}^{J'}$$

が存在して,

$$\sup_{u\in K}\|\mathcal{G}(u)-E\circ\psi\circ F(u)\|_Y<\varepsilon$$

が成り立つ.

補題 3.5 により, 連続作用素 $\mathcal{G}$ はコンパクト集合 K 上で

$$X\xrightarrow{F}\mathbb{R}^J\xrightarrow{\psi}\mathbb{R}^{J'}\xrightarrow{E}Y$$

という有限次元の写像を通じて近似できる. ここで, 本質的な非線形性は有限次元写像 ψ に集約されている. この ψ に対して有限次元の万能近似定理を適用すれば, 通常のニューラルネットワークによる近似が得られる. 一方, 線形写像 F と E は有限ランク作用素や積分作用素として表されるため, これらを組み合わせることで最終的にニューラル作用素 Γ_ε が構成される. この意味で, 作用素版の万能近似定理は, 有限次元版の万能近似定理を基礎として得られるものである.

もっとも, この有限次元化は万能近似性を与える一方で, 近似効率の観点からは重大な問題を残す. 実際, ε を小さくするにつれて, 有限次元化に必要な次元 J, J' 自体が増大しうる. その上で有限次元ニューラルネットワークを適用しても, 次元の呪いにより必要なパラメータ数が大きくなる可能性がある. これが無限次元における「パラメータ複雑性の呪い」につながる.

3.3 無限次元の場合：パラメータ複雑性の呪い

ここでは, Lanthaler-Stuart [31] による, 一般作用素に対する複雑性下界の結果を述べる. X を無限次元 Banach 空間とし, $K\subset X$ が無限次元ハイパーキューブ

$$Q_\alpha:=\left\{u=\sum_{j=1}^{\infty}j^{-\alpha}y_je_j:\ y_j\in[0,1]\right\}\subset X$$

を含むとする. ここで, $\alpha>1$, $\{e_j\}_{j\ge1}$ は X の正規化された基底である.

また, Banach 空間 X 上の汎関数 $\Gamma:X\to\mathbb{R}$ が

$$\Gamma(u)=\Phi(Lu)$$

と表されるとき, これをニューラルネットワーク型汎関数 (neural network-type functional) と呼ぶ. ここで, $J\in\mathbb{N}$, $L:X\to\mathbb{R}^J$ は連続線形写像, $\Phi:\mathbb{R}^J\to\mathbb{R}$ は有限次元 ReLU ニューラルネットワークである. このクラスは, 有限個の線形観測を通して入力関数を有限次元化し, その後に通常のニューラルネットワークを適用するという構造をもつため, DeepONet, PCA-Net, FNO など多くの作用素学習モデルの基本的特徴を抽象化したものとみなせる.

定理 3.6 (Lanthaler-Stuart [31])**.** 上の設定のもとで, 任意の $r \in \mathbb{N}$ および任意の $\delta > 0$ に対して, ある r 回 Fréchet 微分可能な汎関数

$$\Gamma^+ : K \to \mathbb{R}$$

が存在し, 次が成り立つ. 任意の $\varepsilon \in (0,1)$ に対し, あるニューラルネットワーク型汎関数 Γ が

$$\sup_{u \in K} |\Gamma^+(u) - \Gamma(u)| \le \varepsilon$$

を満たすならば, その複雑性は

$$\mathrm{cmplx}(\Gamma) \ge \exp\bigl(c\,\varepsilon^{-1/((\alpha+1+\delta)r)}\bigr)$$

を満たす. ここで, $\mathrm{cmplx}(\Gamma)$ は [31] で導入された複雑性尺度を表し, $c > 0$ は α, r, δ のみに依存する定数である.

定理 3.6 は, 滑らかな作用素であっても, 効率的な近似が本質的に困難なものが存在することを示している. これは有限次元における「次元の呪い」の無限次元版とみなすことができ, 問題はさらに深刻である. 実際, 有限次元では必要な複雑性の下界は典型的には多項式的であったのに対し, 無限次元では指数的下界が現れる.

4　ニューラル作用素の定量的近似定理

本節では, [15] で示された, ニューラル作用素による非線形放物型方程式の解作用素の定量的近似定理を解説する.

前節で見たように, 一般の連続作用素に対する万能近似性だけからは, 効率的な近似は導けず, 無限次元ではパラメータ複雑性の呪いが現れる. これを回避する自然な方針は, 近似対象を PDE の解作用素のような特別な構造をもつ作用素へと制限し, その構造をモデルや学習に取り込むことである. 近年, モデル複雑性の指数的増大を伴わずに, 特定の PDE の解作用素に対する定量的近似定理がいくつか確立されている [11, 15, 26, 29, 32, 39]. 本節で扱う [15] の特徴は, Duhamel 原理と Picard 反復に整合するようにニューラル作用素を構成し, その層構造を解の反復構造と対応づける点にある. これにより, パラメータ複雑性の呪いを回避しうる定量的近似定理が得られる. 本アプローチは, 放物型方程式に限らず, Picard 反復によって解を構成できる他の方程式群にも拡張できる可能性をもつ. なお, 作用素学習の定量的近似に関するその他の方向性については [28] を参照されたい.

4.1 近似対象：非線形放物型方程式の解作用素

本節では, 近似対象の作用素を明確化する. 次の非線形放物型方程式の初期値問題を考える.

$$\begin{cases} \partial_t u + \mathcal{L}u = F(u) & \text{in } (0,T)\times D, \\ u(0) = u_0 & \text{in } D. \end{cases} \tag{P}$$

ここで, $T \in (0,\infty]$, $D \subset \mathbb{R}^d$ は有界領域とし, $u : (0,T)\times D \to \mathbb{R}$ は未知関数, $u_0 : D \to \mathbb{R}$ は初期値とする. 本節を通して, 主要部の作用素 L は半群 $\{S_{\mathcal{L}}(t)\}_{t\geq 0}$ を生成し, さらに各 $t>0$ に対して $S_{\mathcal{L}}(t)$ は積分核 $G(t,x,y)$ をもつと仮定する. また, ある $\nu>0$, $C_{\mathcal{L}}>0$ が存在して, 任意の $1 \leq q_1 \leq q_2 \leq \infty$ に対して,

$$\|S_{\mathcal{L}}(t)\|_{L^{q_1}\to L^{q_2}} \leq C_{\mathcal{L}} t^{-\nu(\frac{1}{q_1}-\frac{1}{q_2})}, \quad t \in (0,1] \tag{H1}$$

が成り立つとする. さらに, 非線形項 $F \in C^1(\mathbb{R};\mathbb{R})$ は $F(0)=0$ を満たし, ある $p>1$, $C_F>0$ が存在して

$$|F(z_1)-F(z_2)| \leq C_F \max_{i=1,2}|z_i|^{p-1}|z_1-z_2|, \quad z_1,z_2 \in \mathbb{R}. \tag{H2}$$

が成り立つとする.

注意 4.1. (P) は抽象的な初期値境界値問題である ($\mathcal{L}$ に境界条件が含まれる). 例えば, $\mathcal{L}$ を Dirichlet 境界条件を伴うラプラシアンとすれば, (P) は Dirichlet 境界値問題となる. $\mathcal{L}$ の典型例として, 境界条件を伴うラプラシアン (Dirichlet, Neumann, Robin 境界条件), Schrödinger 作用素, 楕円型作用素, 高階または分数階ラプラシアンなどが挙げられる. 詳細は [6, 15, 21, 43] などを参照されたい. また, F の典型例は $F(u) = \pm|u|^{p-1}u, \pm|u|^p$ などが挙げられる.

Duhamel 原理より, (適切な条件下で) (P) は次の積分方程式と同値である.

$$u(t) = S_{\mathcal{L}}(t)u_0 + \int_0^t S_{\mathcal{L}}(t-\tau)F(u(\tau))\,d\tau. \tag{P'}$$

命題 4.2 (時間局所適切性)**.** (H1) かつ (H2) を仮定する. r,s は次を満たす.

$$r,s \in [p,\infty], \quad \frac{\nu}{s}+\frac{1}{r} < \frac{1}{p-1}.$$

このとき, 任意の初期値 $u_0 \in L^\infty(D)$ に対し, ある時刻 $T=T(u_0) \in (0,1]$ が存在し, (P') は一意解 $u \in L^r(0,T;L^s(D))$ をもつ. さらに, u は初期値連続依存性を満たす.

証明はスケール劣臨界の場合における通常の Banach の不動点定理を用いた議論に基づく. より正確には, $u_0 \in L^\infty(D)$ かつ $T,M>0$ に対し, 写像 $\Phi = \Phi_{u_0}$ を以下で定める.

$$\Phi[u](t) := S_{\mathcal{L}}(t)u_0 + \int_0^t S_{\mathcal{L}}(t-\tau)F(u(\tau))\,d\tau, \quad t \in [0,T]. \tag{1}$$

さらに, $\mathcal{X} := B_{L^r(0,T;L^s(D))}(M) = \{u \in L^r(0,T;L^s(D)) : \|u\|_{L^r(0,T;L^s(D))} \le M\}$, $\mathrm{d}(u,v) := \|u-v\|_{L^r(0,T;L^s)}$ とすると, $(\mathcal{X},\mathrm{d})$ は完備距離空間である. このとき, T を十分小さくとれば, $\Phi : \mathcal{X} \to \mathcal{X}$ が縮小写像となるため, Banach の不動点定理より, 不動点 $u \in \mathcal{X}$ (i.e. $u = \Phi[u]$) が得られる. これがまさしく (P') の解である. さらに, 次の結果も得られる.

系 4.3 (Picard 反復, [54])**.** 命題 4.2 と同じ仮定を課す. このとき, 任意の $R, M > 0$ と $\delta \in (0,1)$ に対し, 時刻 $T = T(R,M,\delta) \in (0,1]$ が存在し, 次が成り立つ.

(i) $\Gamma^+ : B_{L^\infty}(R) \to B_{L^r(0,T;L^s)}(M)$ が一意的に存在し,

$$\Gamma^+(u_0) = u, \quad u_0 \in B_{L^\infty}(R).$$

ここで, u は命題 4.2 における (P') の解である.

(ii) $u_0 \in B_{L^\infty}(R)$ に対し, Picard 反復を以下で定める.

$$\begin{cases} u^{(1)} := S_{\mathcal{L}}(t)u_0, \\ u^{(\ell+1)} := \Phi[u^{(\ell)}] = u^{(1)} + \displaystyle\int_0^t S_{\mathcal{L}}(t-\tau)F(u^{(\ell)}(\tau))\,d\tau, \quad \ell = 1,2,\cdots. \end{cases}$$

このとき, $u^{(\ell)} \to u \ in\ L^r(0,T;L^s(D))\ as\ \ell \to \infty$ かつ

$$\mathrm{d}(u^{(\ell)},u) \le \frac{\delta^\ell}{1-\delta}\mathrm{d}(u^{(1)},0), \quad \ell = 1,2,\cdots.$$

本節の最後に, Green 関数とその近似を与える. (H1) より $S_{\mathcal{L}}(t)$ の積分核 $G = G(t,x,y)$ (Green 関数) が存在する:

$$S_{\mathcal{L}}(t)u_0(x) = \int_D G(t,x,y)u_0(y)\,dy.$$

便宜上, $t \le 0$ に対して $G(t,x,y) \equiv 0$ と定める. G の基底展開を考え, 近似積分核 G_N を定める.

$$G(t-\tau,x,y) = \sum_{m,n\in\Lambda} c_{m,n}\psi_m(\tau,y)\varphi_n(t,x),$$

$$G_N(t-\tau,x,y) := \sum_{m,n\in\Lambda_N} c_{m,n}\psi_m(\tau,y)\varphi_n(t,x).$$

ただし, $\varphi = \{\varphi_n\}_{n\in\Lambda} \subset L^r(0,T;L^s(D))$, $\psi = \{\psi_n\}_{n\in\Lambda} \subset L^{r'}(0,T;L^{s'}(D))$ とし, $\Lambda_N \subset \Lambda$, $|\Lambda_N| = N$ とする. r', s' はそれぞれ r, s の Hölder 共役指数である.

本稿では, (P') の解作用素 $\Gamma^+ : L^\infty(D) \to L^r(0,T;L^s(D)) : u_0 \mapsto u$ を近似対象の作用素とし, Picard 反復の観点からニューラル作用素による近似定理を確立する.

4.2　近似モデル：ニューラル作用素

本節では, ニューラル作用素の定義を与える. ここで用いるニューラル作用素は, 入力層が線形半群 $S_{\mathcal{L}}(t)$ の近似に対応し, 各中間層が Picard 反復の 1 ステップ Φ に対応するように構成される. 以下では, そのための一般的な枠組みを定義する. 記号 $\langle\cdot,\cdot\rangle$ は $L^s(D)$ と $L^{s'}(D)$ または $L^r(0,T;L^s(D))$ と $L^{r'}(0,T;L^{s'}(D))$ の間の双対積を表すものとする. すなわち,

$$\langle u,v\rangle := \int_D u(x)\,v(x)\,dx \quad \text{あるいは} \quad \langle u,v\rangle := \int_0^T\int_D u(t,x)\,v(t,x)\,dx\,dt.$$

定義 4.4. $T>0$ および $r,s\in[1,\infty]$ とする. $\varphi:=\{\varphi_n\}_n$ および $\psi:=\{\psi_m\}_m$ をそれぞれ $L^r(0,T;L^s(D))$ および $L^{r'}(0,T;L^{s'}(D))$ に属する関数族とする. このとき, ニューラル作用素 Γ を以下で定義する:

$$\Gamma : L^\infty(D)\to L^r(0,T;L^s(D)) : u_0\mapsto \hat{u}^{(L+1)}.$$

ここで, 出力関数 $\hat{u}^{(L+1)}$ は以下の手順によって定義される.

1. **（入力層）** $\hat{u}^{(1)}=(\hat{u}_1^{(1)},\hat{u}_2^{(1)},\dots,\hat{u}_{d_1}^{(1)})$ を以下で定める:

$$\hat{u}^{(1)}(t,x) := (K_N^{(0)}u_0)(t,x)+b_N^{(0)}(t,x).$$

ここで, $K_N^{(0)}:L^\infty(D)\to L^r(0,T;L^s(D))^{d_1}$ および $b_N^{(0)}\in L^r(0,T;L^s(D))^{d_1}$ は次で与えられる:

$$(K_N^{(0)}u_0)(t,x) := \sum_{m,n\in\Lambda_N} C_{n,m}^{(0)}\langle\psi_m(0,\cdot),u_0\rangle\varphi_n(t,x) \quad \text{with } C_{n,m}^{(0)}\in\mathbb{R}^{d_1\times 1},$$
$$b_N^{(0)}(t,x) := \sum_{n\in\Lambda_N} b_n^{(0)}\varphi_n(t,x) \quad \text{with } b_n^{(0)}\in\mathbb{R}^{d_1}.$$

2. **（中間層）** $2\le\ell\le L$ に対して, $\hat{u}^{(\ell)}=(\hat{u}_1^{(\ell)},\hat{u}_2^{(\ell)},\dots,\hat{u}_{d_\ell}^{(\ell)})$ を次で定義する:

$$\hat{u}^{(\ell+1)}(t,x)=\sigma\left(W^{(\ell)}\hat{u}^{(\ell)}(t,x)+(K_N^{(\ell)}\hat{u}^{(\ell)})(t,x)+b^{(\ell)}\right),\quad 1\le\ell\le L-1$$

3. **（出力層）** $\hat{u}^{(L+1)}$ を次で定義する:

$$\hat{u}^{(L+1)}(t,x)=W^{(L)}\hat{u}^{(L)}(t,x)+(K_N^{(L)}\hat{u}^{(L)})(t,x)+b^{(L)}.$$

ここで, $\sigma:\mathbb{R}\to\mathbb{R}$ は各成分ごとに作用する非線形活性化関数であり, $W^{(\ell)}\in\mathbb{R}^{d_{\ell+1}\times d_\ell}$ は ℓ 番目の中間層における重み行列, $b^{(\ell)}\in\mathbb{R}^{d_{\ell+1}}$ はバイアスである。また, $K_N^{(\ell)}:L^r(0,T;L^s(D))^{d_\ell}\to L^r(0,T;L^s(D))^{d_{\ell+1}}$ は次で定義される:

$$(K_N^{(\ell)}u)(t,x) := \sum_{m,n\in\Lambda_N} C_{m,n}^{(\ell)}\langle\psi_m,u\rangle\varphi_n(t,x) \quad \text{with } C_{m,n}^{(\ell)}\in\mathbb{R}^{d_{\ell+1}\times d_\ell}.$$

ここで, $u=(u_1,\dots,u_{d_\ell})\in L^r(0,T;L^s(D))^{d_\ell}$ に対し, 次の記法を用いる:

$$\langle \psi_m, u\rangle := (\langle \psi_m, u_1\rangle,\dots,\langle \psi_m, u_{d_\ell}\rangle)\in\mathbb{R}^{d_\ell}.$$

なお, $d_{L+1}=1$ とする.

仮説空間 $\mathcal{NO}^{L,H,\sigma}_{N,\varphi,\psi}$ を, 深さ L, 総ニューロン数 $H=\sum_{\ell=1}^{L} d_\ell$, 活性化関数 σ, ランク N, 関数族 φ,ψ をもつニューラル作用素全体として定義する.

注意 4.5. 学習パラメータは, 重み $W^{(\ell)}$, バイアス $b^{(\ell)}$, および係数 $C^{(\ell)}_{m,n}$ である. また, 実装においては, 双対積 $\langle \psi_m, u\rangle$ の計算を, 数値積分やニューラルネットワークなどを用いて行う必要がある.

注意 4.6. φ,ψ を Fourier 基底または wavelet 基底とすると, Γ はそれぞれ FNO または WNO に対応する. したがって, 上記の定義は FNO や WNO を含む比較的一般的なニューラル作用素の枠組みになっている.

4.3 定量的近似定理

次が [15] の主結果である.

定理 4.7 ([15])**.** 命題 4.2 と同じ仮定を課す. さらに, 次を仮定する.

$$\begin{aligned} E_G(N) &:= \Big\| \|G(t-\tau,x,y)-G_N(t-\tau,x,y)\|_{L^{r'}_\tau(0,1;L^{s'}_y)} \Big\|_{L^r_t(0,1;L^s_x)} \to 0,\\ E'_G(N) &:= \Big\| \|G(t,x,y)-G_N(t,x,y)\|_{L^{s'}_y} \Big\|_{L^r_t(0,1;L^s_x)} \to 0 \qquad (N\to\infty). \end{aligned} \tag{H3}$$

このとき, 任意の $R>0$ に対し, ある $T\in(0,1]$ が存在し, 次が成り立つ. 任意の $\varepsilon>0$ に対し, L,H,N と $\Gamma\in\mathcal{NO}^{L,H,\mathrm{ReLU}}_{N,\varphi,\psi}$ が存在し,

$$\sup_{u_0\in B_{L^\infty}(R)} \|\Gamma^+(u_0)-\Gamma(u_0)\|_{L^r(0,T;L^s)} \le \varepsilon$$

が成り立つ. ここで,

$$L\le C_L(\log(\varepsilon^{-1}))^2,\ H\le C_H\varepsilon^{-1}(\log(\varepsilon^{-1}))^2,\quad E_G(N)+E'_G(N)\le C_G\varepsilon.$$

注意 4.8. (H3) は作用素 $\mathcal{L}$ と関数族 φ,ψ に関する仮定である. 技術的な仮定であるが, いくつかの例で成立することを確認している ([15] の付録 C を参照).

φ,ψ が Fourier 基底または wavelet 基底のとき, $E_G(N), E'_G(N)$ は N に関して多項式オーダーで減衰する. このように, 定理 4.7 は適切に基底関数を選べばパラメータ複雑性の呪いを回避できることを示している.

4.4　定理 4.7の証明の概略

証明の核は, PDE の解がもつ Picard 反復の構造と, ニューラル作用素の層方向の順伝播とを対応づけることにある. 真の解 $\Gamma^+(u_0)$ とニューラル作用素の出力 $\Gamma(u_0)$ の誤差を, (i) Picard 反復を有限段で打ち切ることによる誤差と, (ii) 各反復段階をニューラル作用素で近似することによる誤差に分けて評価する.

1. まず, 系 4.3より, 真の解 $u = \Gamma^+(u_0)$ はPicard 反復列 $\{u^{(\ell)}\}_{\ell \geq 1}$ によって近似できる:

$$u^{(1)} := S_{\mathcal{L}}(t)u_0, \quad u^{(\ell+1)} := \Phi[u^{(\ell)}],$$

$$\|u - u^{(L+1)}\|_{L^r(0,T;L^s(D))} \to 0 \quad (L \to \infty).$$

したがって, まず L を大きく取ることで, 打ち切り誤差を十分小さくできる.

2. 次に, 各 Picard 反復を近似するニューラル作用素側の反復列 $\{\hat{u}^{(\ell)}\}_{\ell \geq 1}$ を構成する. そのために, Green 関数 G を有限ランク近似した核 G_N と, 非線形項 F を近似する一次元ニューラルネットワーク F_{net} を用いて,

$$\hat{u}^{(1)} := S_{\mathcal{L},N}(t)u_0, \quad \hat{u}^{(\ell+1)} := \Phi_{N,\text{net}}[\hat{u}^{(\ell)}]$$

を定める. ここで,

$$(S_{\mathcal{L},N}(t)u_0)(x) := \int_D G_N(t,x,y)u_0(y)\,dy,$$

$$\Phi_{N,\text{net}}[v](t,x) := (S_{\mathcal{L},N}(t)u_0)(x) + \int_0^t \int_D G_N(t-\tau,x,y)\,F_{\text{net}}(v(\tau,y))\,dy\,d\tau.$$

このとき,

$$\|u^{(L+1)} - \hat{u}^{(L+1)}\|_{L^r(0,T;L^s(D))}$$

は, Green 関数の有限ランク近似誤差と非線形項 $F - F_{\text{net}}$ の近似誤差を用いて評価できる. 前者は仮定 (H3) により制御され, 後者には一次元 ReLU ニューラルネットワークの定量的近似定理 [17] を適用する.

$$u_0 \xrightarrow{S_{\mathcal{L}}} u^{(1)} \xrightarrow{\Phi} u^{(2)} \xrightarrow{\Phi} \cdots \xrightarrow{\Phi} u^{(L+1)} (\cdots \xrightarrow{L\to\infty} u) \qquad \text{(Picard 反復)}$$

$$u_0 \xrightarrow{S_{\mathcal{L},N}} \hat{u}^{(1)} \xrightarrow{\Phi_{N,net}} \hat{u}^{(2)} \xrightarrow{\Phi_{N,net}} \cdots \xrightarrow{\Phi_{N,net}} \hat{u}^{(L+1)} \qquad \text{(ニューラル作用素の順伝播)}$$

3. 最後に, 写像

$$u_0 \mapsto \hat{u}^{(L+1)}$$

が定義 4.4 の意味でニューラル作用素として実現できることを示す. 具体的には, $\hat{u}^{(L+1)} = \Phi_{N,\mathrm{net}}^{[L]} \circ S_{\mathcal{L},N}[u_0]$ は次のように表現できる:

$$\begin{aligned}\Gamma(u_0) &:= \Phi_{N,\mathrm{net}}^{[L]} \circ S_{\mathcal{L},N}[u_0] \\ &= \tilde{W}' \circ \Big[(\tilde{W} + \tilde{K}_N) \circ \tilde{F}_{\mathrm{net}} \circ \cdots \circ (\tilde{W} + \tilde{K}_N) \circ \tilde{F}_{\mathrm{net}}\Big] \circ (\tilde{K}_N^{(0)} + \tilde{b}_N^{(0)})(u_0).\end{aligned}$$

括弧 $[\circ \cdots \circ]$ 内の合成は $(\tilde{W} + \tilde{K}_N) \circ \tilde{F}_{\mathrm{net}}$ が L 回繰り返される. ここで, $\tilde{W}', \tilde{W}$ はある重み行列, $\tilde{K}_N, \tilde{K}_N^{(0)}$ はある非局所作用素, $\tilde{b}_N^{(0)}$ はあるバイアス関数, $\tilde{F}_{\mathrm{net}}$ はあるニューラルネットワークである. 上記の Γ はニューラル作用素の定義 4.4 の特別な形になっている. ここで重要なのは, 真の解を構成する反復構造がそのままモデルの層構造に埋め込まれる点である.

以上を合わせると,

$$\|\Gamma^{+}(u_0) - \Gamma(u_0)\| \le \|u - u^{(L+1)}\| + \|u^{(L+1)} - \hat{u}^{(L+1)}\|$$

が得られ, 第 1 項は $L \to \infty$ で, 第 2 項は $N \to \infty$ および $F_{\mathrm{net}} \to F$ により小さくできる. これにより, 定理 4.7 の結論が従う.

本結果は, 一般の連続作用素に対する有限次元化と万能近似定理の直接適用によるのではなく, PDE の解がもつ反復構造とニューラル作用素の層構造を対応させることによって, 効率的な定量的近似を実現している. 厳密な証明は [15] の付録 D を参照されたい.

5 議論と今後の展望

最後に, 主結果に関するいくつかの考察と今後の展望について述べる.

(1) **適用可能性と限界.** 本稿で扱ったのは, 冪型非線形項をもつ非線形放物型方程式である. しかし, 本結果の本質は放物型であることそのものではなく, 解作用素が Duhamel 型の積分方程式として表され, しかもその解が Banach の不動点定理に基づく Picard 反復によって構成できる点にある. この意味で, 本稿の枠組みはより広いクラスの方程式へ拡張できる可能性をもつ. 一方, p-Laplacian 方程式のように主要部そのものに非線形性を含む方程式では, このような積分方程式表示を得ることが一般には容易でないため, 本稿の方法をそのまま適用することは難しいと考えられる.

(2) **定理の拡張.** 本稿では, 局所適切性および近似誤差評価を Lebesgue 空間の枠組みで述べた. しかし, PDE 理論の観点からは, Sobolev 空間や Besov 空間など, より精密な関数空間の枠組みで結果を定式化することが自然である. また, 学習理論の観点からも, 解そのものだけでなく, 偏導関数も含めて適切に近似できることを保証するのは重要である. したがって, 本稿の主定理を Sobolev 空間や他の関数空間の枠組みに拡張することは, 今後の重要な課題である.

(3) **長時間解の近似.** 方程式 (P) が時間大域的に適切であるとき, 時間局所解作用素を各時間区間ごとに繰り返し適用することにより, 解をより長い時間区間へ延長することができる. したがって, 各局所時間区間で Γ が Γ^{+} を十分よく近似するならば, その反復により長時間解も段階的に近似できると期待される. このような時間大域的な解の学習についても, 今後さらに検討すべき課題である.

(4) **実装への含意.** 本稿では近似理論に焦点を当て, 実装や数値実験には立ち入らなかった. しかし, 本稿の証明は構成的であり, ニューラル作用素の設計に対して明確な指針を与えている. 実際, ニューラル作用素を

$$\Gamma(u_0) = \widetilde{W}' \circ \left[(\widetilde{W} + \widetilde{K}_N) \circ \widetilde{F}_{\mathrm{net}}\right]^{[J-1]} \circ (\widetilde{K}_N^{(0)} + \widetilde{b}_N^{(0)})(u_0)$$

のように表すと, $(\widetilde{W} + \widetilde{K}_N) \circ \widetilde{F}_{\mathrm{net}}$ が Picard 反復の 1 ステップに対応していることが分かる. この意味で, 本稿の構成は, 同一のブロックを繰り返し用いる反復型アーキテクチャに近い構造をもっている.

一方, 本稿で構成したニューラル作用素は各層に双対積を含むため, そのままでは計算コストが高くなる可能性がある. この点については, 例えば時間方向の Fourier 変換を用いた FNO の実装技術 [27] などが有用な手がかりを与えると期待される. 具体的な実装や実験的検証については今後の課題である.

(5) **構造保存性をもつニューラル作用素.** 数値解析においては, 離散化された近似解が元の方程式のもつ本質的性質を保つことが重要である. 同様に, ニューラル作用素を PDE ソルバーの代理モデルとして用いる以上, 学習されたニューラル作用素が元の解作用素のもつ性質をどの程度保っているかを問う必要がある. どのような構造保存性を課すべきか, またそれをどのように理論的に保証するかは, 今後の重要課題である.

本稿の結果は, ニューラル作用素が単に一般の連続作用素を近似できるだけでなく, PDE の解作用素がもつ積分表示, 不動点構造, Picard 反復といった数理的構造を活用することで, 効率的な近似器として機能しうることを示している. このことは, 作用素学習の理論

において, 近似対象の構造をいかにモデルに取り込むかが本質的に重要であることを示唆している. 今後は, より広い PDE への拡張, 実装と数値検証, さらに構造保存性を備えたニューラル作用素の構成を通じて, 作用素学習と PDE 理論との結びつきがいっそう深まることが期待される.

参考文献

[1] A. R. Barron, *Universal Approximation Bounds for Superpositions of a Sigmoidal Function*, IEEE Transactions on Information Theory **39** (1993), 930–945.

[2] P. Berens, K. Cranmer, N. D. Lawrence, U. von Luxburg, and J. Montgomery, *AI for Science: An Emerging Agenda*, arXiv preprint arXiv:2303.04217, 2023.

[3] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model Reduction and Neural Networks for Parametric PDEs*, The SMAI Journal of Computational Mathematics **7** (2021), 121–157.

[4] B. Bonev, T. Kurth, C. Hundt, J. Pathak, M. Baust, K. Kashinath, and A. Anandkumar, *Spherical Fourier Neural Operators: Learning Stable Dynamics on the Sphere*, in *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research **202** (2023), 2806–2823.

[5] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems*, Proceedings of the National Academy of Sciences of the United States of America **113** (2016), 3932–3937.

[6] T. A. Bui, P. D'Ancona, and F. Nicola, *Sharp L^p estimates for Schrödinger groups on spaces of homogeneous type*, Revista Matemática Iberoamericana **36** (2020), no. 2, 455–484.

[7] T. Chen and H. Chen, *Approximations of continuous functionals by neural networks with application to dynamic systems*, IEEE Transactions on Neural Networks **4** (1993), no. 6, 910–918.

[8] T. Chen and H. Chen, *Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems*, IEEE Transactions on Neural Networks **6** (1995), 911–917.

[9] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural Ordinary Differential Equations*, in *Advances in Neural Information Processing Systems* **31** (2018), 6571–6583.

[10] G. Chen, X. Liu, Q. Meng, L. Chen, C. Liu, and Y. Li, *Learning Neural Operators on Riemannian Manifolds*, National Science Open **3** (2024), 20240001.

[11] K. Chen, C. Wang, and H. Yang, *Deep Operator Learning Lessens the Curse of Dimensionality for PDEs*, Transactions on Machine Learning Research (2023).

[12] G. Cybenko, *Approximation by Superpositions of a Sigmoidal Function*, Mathematics of Control, Signals, and Systems **2** (1989), 303–314.

[13] R. DeVore, B. Hanin, and G. Petrova, *Neural Network Approximation*, Acta Numerica **30** (2021), 327–444.

[14] F. Dietrich and W. Schilders, *Scientific machine learning*, Mathematische Semesterberichte **72** (2025), no. 2, 89–115.

[15] T. Furuya, K. Taniguchi, and S. Okuda, *Quantitative Approximation for Neural Operators in Nonlinear Parabolic Equations*, in *Proceedings of the International Conference on Learning Representations*, 2025.

[16] K.-I. Funahashi, *On the Approximate Realization of Continuous Mappings by Neural Networks*, Neural Networks **2** (1989), 183–192.

[17] I. Gühring, G. Kutyniok, and P. Petersen, *Error Bounds for Approximations with Deep ReLU Neural Networks in $W^{s,p}$ Norms*, Analysis and Applications **18** (2020), 803–859.

[18] G. Gupta, X. Xiao, and P. Bogdan, *Multiwavelet-Based Operator Learning for Differential Equations*, in *Advances in Neural Information Processing Systems* **34** (2021), 24048–24062.

[19] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Networks **4** (1991), 251–257.

[20] S. Huang, W. Feng, C. Tang, Z. He, C. Yu, and J. Lv, *Partial Differential Equations Meet Deep Neural Networks: A Survey*, IEEE Transactions on Neural Networks and Learning Systems **36** (2025), no. 8, 13649–13669.

[21] M. Ikeda, K. Taniguchi, and Y. Wakasugi, *Global existence and asymptotic behavior for semilinear damped wave equations on measure spaces*, Evolution Equations and Control Theory **13** (2024), no. 4, 1101–1125.

[22] H. Jaeger, *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks*, GMD Report 148, German National Research Center for Information Technology, 2001.

[23] H. Jung, J. Gupta, B. Jayaprakash, M. Eagon, H. P. Selvam, C. Molnar, W. Northrop, and S. Shekhar, *A Survey on Solving and Discovering Differential Equations Using Deep Neural Networks*, arXiv preprint arXiv:2304.13807, 2023.

[24] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, *Physics-Informed Machine Learning*, Nature Reviews Physics **3** (2021), 422–440.

[25] P. Kidger and T. Lyons, *Universal approximation with deep narrow networks*, in *Proceedings of the 33rd Annual Conference on Learning Theory*, Proceedings of Machine Learning Research **125** (2020), 2306–2327.

[26] N. Kovachki, S. Lanthaler, and S. Mishra, *On Universal Approximation and Error Bounds for Fourier Neural Operators*, Journal of Machine Learning Research **22** (2021), 1–76.

[27] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Neural Operator: Learning Maps Between Function Spaces with Applications to PDEs*, Journal of Machine Learning Research **24** (2023), 1–97.

[28] N. B. Kovachki, S. Lanthaler, and A. M. Stuart, *Operator Learning: Algorithms and Analysis*, Handbook of Numerical Analysis **25** (2024), 419–467.

[29] S. Lanthaler, *Operator Learning with PCA-Net: Upper and Lower Complexity Bounds*, Journal of Machine Learning Research **24** (2023), 1–67.

[30] S. Lanthaler, S. Mishra, and G. E. Karniadakis, *Error Estimates for DeepONets: A Deep Learning Framework in Infinite Dimensions*, Transactions of Mathematics and Its Applications **6** (2022), tnac001.

[31] S. Lanthaler and A. M. Stuart, *The Parametric Complexity of Operator Learning*, IMA Journal of Numerical Analysis **46** (2026), no. 2, 647–712.

[32] S. Lanthaler, Z. Li, and A. M. Stuart, *Nonlocality and Nonlinearity Implies Universality in Operator Learning*, Constructive Approximation **62** (2025), 261–303.

[33] M. Leshno, V. Ya. Lin, A. Pinkus, and S. Schocken, *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*, Neural Networks **6** (1993), 861–867.

[34] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Neural Operator: Graph Kernel Network for Partial Differential Equations*, arXiv preprint arXiv:2003.03485, 2020.

[35] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Fourier Neural Operator for Parametric Partial Differential Equations*, in *Proceedings of the International Conference on Learning Representations*, 2021.

[36] Q. Li, C. Tai, and W. E, *Stochastic Modified Equations and Dynamics of Stochastic Gradient Algorithms I: Mathematical Foundations*, Journal of Machine Learning Research **20** (2019), 1–47.

[37] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators*, Nature Machine Intelligence **3** (2021), 218–229.

[38] W. Maass, T. Natschläger, and H. Markram, *Real-Time Computing without Stable States: A New Framework for Neural Computation Based on Perturbations*, Neural Computation **14** (2002), 2531–2560.

[39] C. Marcati and C. Schwab, *Exponential Convergence of Deep Operator Networks for Elliptic Partial Differential Equations*, SIAM Journal on Numerical Analysis **61** (2023), 1513–1545.

[40] S. Mandt, M. D. Hoffman, and D. M. Blei, *Stochastic Gradient Descent as Approximate Bayesian Inference*, Journal of Machine Learning Research **18** (2017), 1–35.

[41] N. Murata, *An Integral Representation of Functions Using Three-Layered Networks and Their Approximation Bounds*, Neural Networks **9** (1996), no. 6, 947–956.

[42] S. Musslick, L. K. Bartlett, S. H. Chandramouli, et al., *Automating the practice of science: Opportunities, challenges, and implications*, Proceedings of the National Academy of Sciences of the United States of America **122** (2025), no. 5, e2401238121.

[43] E. M. Ouhabaz, *Analysis of Heat Equations on Domains*, Princeton University Press, Princeton, NJ, 2005.

[44] A. Pinkus, *Approximation Theory of the MLP Model in Neural Networks*, Acta Numerica **8** (1999), 143–195.

[45] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations*, Journal of Computational Physics **378** (2019), 686–707.

[46] J. Schmidt-Hieber, *Nonparametric regression using deep neural networks with ReLU activation function*, The Annals of Statistics **48** (2020), 1875–1897.

[47] S. Sonoda, *Integral Representation of Neural Networks and Ridgelet Transform*, Bull. JSIAM **33** (2023), 4–13.

[48] S. Sonoda and N. Murata, *Neural Network with Unbounded Activation Functions is Universal Approximator*, Applied and Computational Harmonic Analysis **43** (2017), no. 2, 233–268.

[49] T. Suzuki, *Adaptivity of Deep ReLU Network for Learning in Besov and Mixed Smooth Besov Spaces: Optimal Rate and Curse of Dimensionality*, in *Proceedings of the International Conference on Learning Representations*, 2019.

[50] M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert, *PDEBench: An Extensive Benchmark for Scientific Machine Learning*, in *Advances in Neural Information Processing Systems* **35** (2022), 1596–1611.

[51] T. Tripura and S. Chakraborty, *Wavelet Neural Operator for Solving Parametric Partial Differential Equations in Computational Mechanics Problems*, Computer Methods in Applied Mechanics and Engineering **404** (2023), 115783.

[52] H. Wang, T. Fu, Y. Du, et al., *Scientific discovery in the age of artificial intelligence*, Nature **620** (2023), no. 7972, 47–60.

[53] D. Yarotsky, *Error Bounds for Approximations with Deep ReLU Networks*, Neural Networks **94** (2017), 103–114.

[54] E. Zeidler, *Nonlinear Functional Analysis and its Applications I: Fixed-Point Theorems*, Springer, New York, 1986.

[55] H. Zhang and D. V. Vargas, *A Survey on Reservoir Computing and Its Interdisciplinary Applications Beyond Traditional Machine Learning*, IEEE Access **11** (2023), 81033–81070.

Koichi Taniguchi
Faculty of Engineering
Shizuoka University
3-5-1 Johoku, Chuo-ku, Hamamatsu, 432-8561
JAPAN
E-mail address: taniguchi.koichi@shizuoka.ac.jp