ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2023, том 514, № 2, с. 72–79

УДК 004.8

СПЕКТРАЛЬНЫЕ НЕЙРОННЫЕ ОПЕРАТОРЫ

© 2023 г. В. С. Фанасков[1,*], И. В. Оселедец[1,2,]**

Представлено академиком РАН А.А. Шананиным
Поступило 30.08.2023 г.
После доработки 06.09.2023 г.
Принято к публикации 15.10.2023 г.

В недавних работах были построены нейронные операторы – обобщение нейронных сетей, позволяющее приближать отображения между бесконечномерными векторными пространствами. Используя численные и аналитические методы, мы продемонстрируем особенности обучения и оценки эффективности для подобных нейронных сетей. В частности, мы покажем, что для широкого класса нейронных операторов, основанных на интегральных преобразованиях, неизбежно появление систематической ошибки, связанной с наложением спектра. Чтобы избежать этой ошибки, мы введем *спектральные нейронные операторы*, которые в явном виде используют дискретизацию для областей определения и значения. Несмотря на то что дискретизация уменьшает способность к аппроксимации, на практике оказывается, что спектральные нейронные операторы часто превосходят по точности нейронные операторы, определенные на бесконечномерных банаховых пространствах.

Ключевые слова: нейронные операторы, уравнения в частных производных

DOI: 10.31857/S2686954323601422, **EDN:** AEOEQY

1. ВВЕДЕНИЕ

Численное уравнение в частных производных имеет множество практических приложений. Это приводит к тому, что регулярно появляются новые схемы поиска приближенного решения. Недавно к традиционным схемам (конечно-разностным, спектральным, конечно-элементным) добавились статистические подходы на основе машинного обучения.

В качестве примера можно привести схемы получения решения, основанные на прямой минимизации невязки [4], сверточных сетях [12], выучивании оптимальной дискретизации [9] и нейронных операторах [10, 13]. Последний подход представляет наибольший интерес благодаря хорошей точности и небольшой вычислительной сложности.

Нейронный оператор представляет собой нейронную сеть, которая принимает на вход функции $a_i(x_h)$, $i = 1,\ldots,M_1$, вычисленные на сетке x_h, и возвращает функции $b_j(x_h)$, $j = 1,\ldots,M_2$. При этом для сеток с разной дискретизацией h_1 и h_2 результат $b_j(x_{h_1}) \simeq b_j(x_{h_2})$ оказывается одной и той же функцией. Это свойство называют “инвариантность относительно дискретизации”. Позже, в Разделе 2, мы продемонстрируем, что для широкого класса нейронных операторов это свойство выполняется только приближенно.

Как правило, задача для нейронного оператора $N(u,x;w)$ ставится как поиск параметров w, которые дают наилучшее приближение для оператора решения уравнения в частных производных:

$$w = \arg\min_{\tilde{w}} \mathbb{E}_{u\sim P}\left[\left\|N(u,x;\tilde{w}) - t[u]\right\|\right], \qquad (1)$$

где w, $\tilde{w}$ – параметры нейронного оператора, u – входные данные для уравнения в частных производных из распределения P, t – оператор решения уравнения в частных производных, $\|\cdot\|$ – подходящая функциональная норма, например L_2.

Наиболее часто задача (1) решается в два этапа. На первом при помощи классического метода решения составляется выборка пар $u, t[u]$. На втором используются методы градиентного спуска для поиска локально оптимальных параметров w.

Одна из ключевых особенностей (1) состоит в том, что P – это распределение на пространстве функций. С другой стороны, способ получения приближенных решений, на которых оператор обучается, все равно подразумевает дискретизацию, поэтому вопрос о необходимости непрерыв-

[1]*Сколковский институт науки и технологий, Москва, Россия*
[2]*AIRI, Москва, Россия*
**E-mail: v.fanaskov@skoltech.ru*
***E-mail: i.oseledets@skoltech.ru*

ной формулировки остается открытым. В Разделе 2 мы продемонстрируем, что дискретизация может приводить к тому, что обученный нейронный оператор дает хорошее приближение только на конечномерных подпространствах.

Имея в виду неточность инвариантности относительно дискретизации и эффективную конечномерность нейронных операторов, естественно рассмотреть явную дискретизацию для выходных данных u и решения $t[u]$. В Разделе 3 мы введем конечномерный спектральный нейронный оператор, обладающий инвариантностью относительно дискретизации по построению. Явная дискретизация потенциально уменьшает точность, поэтому в Разделе 4 мы проведем численное сравнение различных нейронных операторов.

В заключении введения вы приводим краткое содержание основных результатов статьи:

1. В Разделе 2 собраны результаты о способности нейронных операторов к обобщению и ошибках, связанных с наложением спектра. Показано, что нейронные операторы выучивают отображения между конечномерными пространствами. Также продемонстрирован эффект наложения спектра, благодаря которому при измельчении сетки точность нейронного оператора падает.

2. В Разделе 3 описан спектральный нейронный оператор.

3. Раздел 4 на широком классе уравнений продемонстрировало, что точность спектрального оператора часто превосходит точность нейронных операторов.

2. СВОЙСТВА НЕЙРОННЫХ ОПЕРАТОРОВ

На данный момент существуют десятки различных нейронных операторов, среди которых можно выделить метаархитектуры (например, глубокие операторные сети [10] (DeepONet)) и операторы, основанные на интегральных преобразованиях (например, нейронный оператор Фурье (FNO) [13]). Мы сосредоточимся на свойствах сетей с интегральными операторами.

Сети с интегральными операторами основаны на естественном обобщении линейного слоя. Классическая нейронная сеть преобразует векторы при помощи оператора

$$v = Au + b, \quad v, b \in \mathbb{R}^n, \quad u \in \mathbb{R}^m, \quad A \in \mathbb{R}^{n\times m}, \quad (2)$$

где v и u – входной и выходной векторы слоя, A и b – параметры слоя. По аналогии, линейный слой нейронного оператора имеет вид

$$v(x) = \int dy\, A(x, y; w_A)u(y) + b(x; w_b), \quad (3)$$

где $v(x)$ и $u(x)$ – входные и выходные функции слоя, $A(x, y; w_A)$ и ядро интегрального оператора, $w_A \in \mathbb{R}^{N_A}$ и $w_b \in \mathbb{R}^{N_A}$ – параметры слоя.

Из соображений численной эффективности для реализации линейного слоя (3) часто используется параметризация

$$\int dy\, A(x, y; w_A)u(y) = \sum_{ij} \phi_j(x) w_{ji} \underbrace{\int dy \phi_i(y)u(y)}_{c_i}, \quad (4)$$

вместе с быстрыми преобразованиями, которые позволяют эффективно вычислить коэффициенты $c_i = \int dy \phi_i(y)u(y)$ и сумму $\sum_j \phi_j(x)\tilde{c}_j$.

Интегральное ядро (4) используется во многих нейронных операторах, например, в нейронном операторе Фурье (FNO) [13], вейвлетном операторе (WNO) [17] и вейвлетном операторе на основе нестандартного вейвлетного преобразования (MWNO) [15].

Выбор параметризации ядра (4), представления функций $u(y)$ и способа обучения приводит к ряду последствий, которые ставят под вопрос необходимость рассмотрения непрерывной формулировки (над функциональными пространствами) (1). В следующих двух секциях мы рассмотрим два таких эффекта: обобщение за пределами обучающей выборки и ошибки, возникающие из-за наложения спектра.

2.1. Способность к обобщению

Основной особенностью нейронных операторов является возможность работать с функциями, вычисленными на сетках с разным разрешением. Очевидно, на сетках с большим разрешением можно определить более сложные функции. В качестве примера можно привести следующие хорошо известные результаты:

1. Теорема Найквиста-Шеннона о возможности восстановить коэффициенты ряда Фурье из значений функции на равномерной сетке [5, Глава 3].

2. Сходная теорема для полиномов Чебышева и функциях, вычисленных на сетках Чебышева [11, Глава 4].

3. Описание пространств Бесова с использованием вейвлетов [7].

По построению нейронный оператор можно обучить на функциях, посчитанных на сетках с низким разрешением, и затем использовать на сетках с более высоким разрешением. Может ли нейронный оператор успешно экстраполировать с низкого на высокое разрешение?

В нескольких работах [13, 14, 16] авторы положительно отвечают на этот вопрос, утверждая, что ошибка приближенного решения, получен-

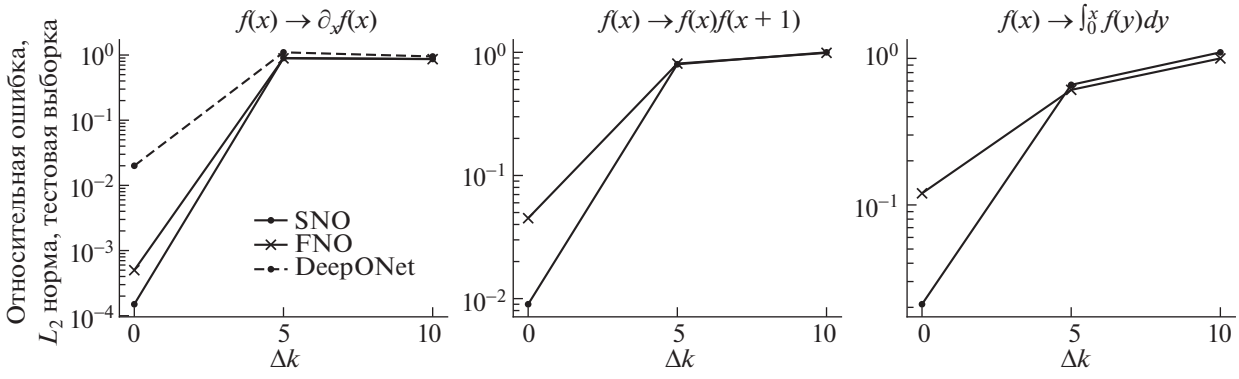

Рис. 1. Демонстрация способности к обобщению на функции, не представленные в обучающей выборке. Для всех примеров нейронные операторы обучались на тригонометрических полиномах (8) с гармониками $k_{\min} = 15$, $k_{\max} = 25$. В качестве метрики используется относительная ошибка $\|f(x) - f_{\text{appr}}(x)\|_2 / \|f(x)\|_2$. Для неопределенного интеграла (6) тестовая выборка состоит из тригонометрических полиномов с $k_{\min} = 15 - \Delta k$, $k_{\max} = 25 - \Delta k$. Для операторов (7) и (5) тестовую выборку составляют тригонометрические полиномы с $k_{\min} = 15 + \Delta k$, $k_{\max} = 25 + \Delta k$.

ного нейронным оператором, *не увеличивается* при увеличении разрешения сетки. Однако во всех предоставленных примерах сетки были выбраны таким способом, что все функции хорошо приближались на сетках с низким разрешением, поэтому увеличение разрешения не приводило к усложнению функции.

Чтобы в явном виде проконтролировать способность нейронных операторов к обобщению, мы обучили нейронные операторы приближать три простых оператора

$$f(x) \to f(x)f(x+1), \tag{5}$$

$$f(x) \to \int_0^x dy\, f(y), \tag{6}$$

$$f(x) \to \frac{df(x)}{dx}, \tag{7}$$

действующих на семействе тригонометрических полиномов

$$f(x) = \mathcal{R}\left[\sum_{k=k_{\min}}^{k_{\max}} d_k \exp(\pi i k x)\right], \tag{8}$$

где $\mathcal{R}$ – вещественная часть, коэффициенты d_k имеют нормально распределенные вещественную и мнимую части и $k_{\max} > k_{\min}$ выбраны разными для обучающей и тестовой выборки.

Результаты для трех различных операторов представлены на рис. 1. Легко видеть, что нейронные операторы не обобщают за пределами обучающей выборки, даже если рассматриваются простейшие линейные операторы.

2.2. Ошибки наложения спектра

Нейронный оператор состоит из нелокальных интегральных преобразований (4) и поточечного применения активационных функций, например $\mathrm{ReLU}(x) \equiv \sigma(x) = x \max\{0, x\}$. Чтобы продемонстрировать проблемы, возникающие из-за наложения спектра, рассмотрим периодические функции и нейронный оператор Фурье.

Предположим, входная функция $f(x)$ может быть точно представлена рядом Фурье с гармониками $|k| \le N$. Согласно теореме Найквиста-Шеннона можно эквивалентно хранить значения функции на однородной сетке с $2N+1$ точками. Интегральный оператор (4) по построению использует только низкие частоты, поэтому с ним не возникает проблем. Однако поточечные активационные функции $\sigma(f(x))$ вносят высокие частоты. Может оказаться, что эти частоты превышают частоту Найквиста используемой однородной сетки. В этом случае более высокие частоты неотличимы от низких (например, для полиномов Чебышева [9, Chapter 11, Определение 23] и ряда Фурье [11, Глава 4, Теорема 4.1] для ряда Чебышева). Это приводит к ошибкам в вычислении нейронного оператора Фурье.

Для активационной функции σ и сетки с $2N+1$ точками, можно оценить эти ошибки, количественно используя следующую метрику

$$\sigma(f(x)) \equiv \sum_i c_i \phi_i, \quad E_a(f, \sigma) \equiv \frac{\left\| \sum_{|i|>N} c_i \phi_i \right\|}{\left\| \sigma(f(x)) \right\|}. \tag{9}$$

Ошибка (9) позволяет оценить норму гармоник, которые невозможно разрешить на заданной сетке, по отношению к полной норме функции

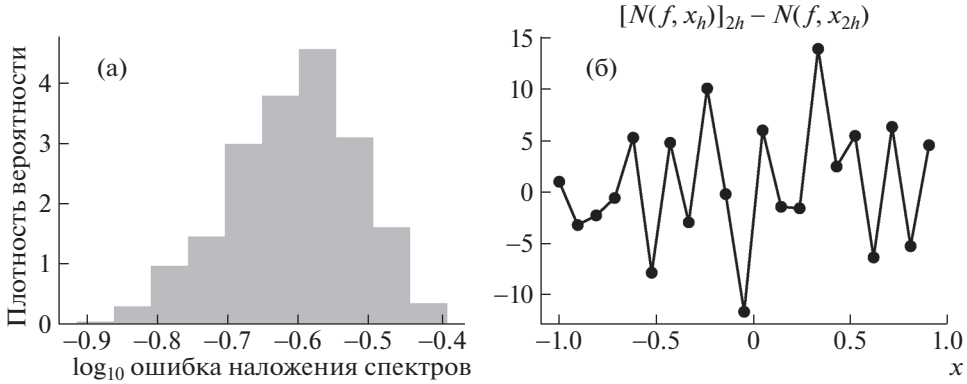

Рис. 2. Оценка ошибки наложения спектра для нейронного оператора Фурье. Результаты для оператора дифференцирования. (а) Распределение логарифма относительной ошибки наложения спектров для 750 входных функций. Средняя относительная ошибка составляет ~25%. (б) Ошибка наложения спектра для конкретной функции.

после применения активации. В качестве примера оценки ошибки, мы приводим точный результат для $f(x)$ специального вида.

Лемма 1 (Наложение спектра для ReLU(x)). *На однородной сетке или сетке Чебышева с* $2N+1$ *точками ошибка наложения спектра* E_a *из* (9) *равна*

$$E_a(\cos(\pi Nx), \mathrm{ReLU}) = E_a(T_N(x), \mathrm{ReLU}) = \\ = \frac{1}{\pi}\sqrt{\frac{\pi^2}{2} - 4} \simeq 0.31. \tag{10}$$

Доказательство. Легко видеть, что

$$\mathrm{ReLU}(x) = \sum_{i=0}^{\infty} p_i T_i(x), \qquad p_i = \begin{cases} \dfrac{1}{\pi}, & i = 0; \\ \dfrac{1}{2}, & i = 1; \\ \dfrac{2}{\pi}\dfrac{\cos(\pi i/2)}{1-i^2}, & i \geq 2, \end{cases} \tag{11}$$

для всех $x \in [-1,1]$. Так как область значений $\cos(\pi Nx)$ и T_N лежит в интервале $[-1, 1]$ мы можем использовать ряд Чебышева вместо ReLU, для вычисления композиции. Используя $T_n(T_m(x)) = T_{nm}(x)$, а также $T_n(\cos(\theta)) = \cos(n\theta)$ получаем

$$\mathrm{ReLU}(T_N(x)) = \sum_{i=0}^{\infty} p_i T_{iN}(x), \\ \mathrm{ReLU}(\cos(\pi Nx)) = \sum_{i=0}^{\infty} p_i \cos(\pi i N x). \tag{12}$$

Используя норму полиномов Чебышева (с весом $1/\sqrt{1-x^2}$) $\|T_i(x)\|^2 = \pi/(2-\delta_{i0})$ и норму тригонометрического полинома $\|\cos(\pi nx)\|_2^2 = (1+\delta_{i0})$, мы получаем для ошибок наложения спектра

$$E_a(\cdot, \mathrm{ReLU}) = \sqrt{\frac{\sum_{i=2}^{\infty} p_i^2}{2p_0^2 + p_1^2 + \sum_{i=2}^{\infty} p_i^2}}. \tag{13}$$

Используя равенство Парсеваля

$$2p_0^2 + p_1^2 + \sum_{i=2}^{\infty} p_i^2 = \int \left(\mathrm{ReLU}(\cos(\pi Nx))\right)^2 dx = \\ = \frac{1}{2}\int \left(\cos(\pi Nx)\right)^2 dx, \tag{14}$$

получаем $\sum_{i=2}^{\infty} p_i^2 = (\pi^2 - 8)/4\pi^2$.

Лемма 1 иллюстрирует, что наложение спектра вносит систематическую ошибку в обучение нейронного оператора Фурье. А именно, во время обучения нейронный оператор, вычисленный на сетке с недостаточным разрешением, учится компенсировать ошибку наложения спектра, которую вносят активационные функции. Когда позднее нейронный оператор вычисляется на сетке высокого разрешения, ошибка наложения спектра отсутствует, но как и на стадии обучения нейронный оператор подавляет эту ошибку, поэтому в предсказания нейронного оператора вносится дополнительная неточность.

Лемма 1 дает оценку только для одного слоя. Чтобы оценить ошибку наложения спектра для всей нейронной сети, мы провели эксперименты для нейронного оператора Фурье для трех операторов (5)–(7). Чтобы измерить ошибку наложения спектра для каждой входной функции, мы посчитали нейронный оператор для двух разрешений и вычислили относительную ошибку

$$\|[N(f, x_h)]_{2h} - N(f, x_{2h})\|_2 \,/\, \|N(f, x_h)\|_2. \tag{15}$$

Результаты и детали экспериментов можно найти на рис. 2 и рис. 3.

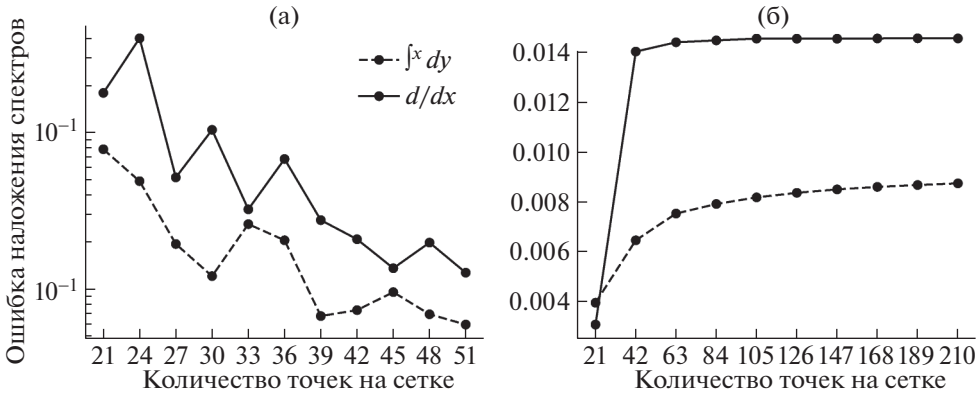

Рис. 3. Оценка ошибки наложения спектра для нейронного оператора Фурье. Результаты для операторов интегрирования и дифференцирования. (а) Относительная ошибка для операторов, обученных на сетках с разным количеством точек. Ошибка уменьшается с отдалением максимальной частоты функции от частоты Найквиста. (б) Относительная ошибка для операторов, обученных на сетке с 21 точкой, вычисленных на сетках с разным количеством точек. Ошибка быстро насыщается с увеличением количества точек на сетке.

Из результатов понятно, что ошибка наложения спектра может быть достаточно большой, если обучение происходит на сетке с недостаточным разрешением. Ошибка ожидаемо уменьшается при увеличении разрешения сетки.

3. СПЕКТРАЛЬНЫЕ НЕЙРОННЫЕ ОПЕРАТОРЫ

В Секции 2 мы продемонстрировали, что нейронные операторы выучивают отображение между конечномерными векторными пространствами. Размерности пространств и базисы зависят от архитектуры нейронной сети и данных, использованных для обучения. Таким образом, естественно рассмотреть дискретизацию пространств определения и значения нейронного оператора:

$$\sum_i g_i(x)d_i = f_{\text{in}}(x) \xrightarrow{\text{N}} f_{\text{out}}(x) = \sum_i g_i(x)b_i, \quad (16)$$

где $g_i(x)$ либо полиномы Чебышева, либо комплексные экспоненты. Мы называем любое отображение вида (26) *спектральным нейронным оператором*. Спектральный нейронный оператор по сути обычная нейронная сеть, определенная на конечномерном векторном пространстве. Входные данные представляют собой вектор коэффициентов $d_i, i = 1,\ldots,M$, а выходные – вектор коэффициентов $b_j, j = 1,\ldots,N$. Так как спектральный оператор определен непосредственно в терминах коэффициентов, явная зависимость от сетки исчезает и проблема наложения спектра не возникает. Интуитивно понятно, что дискретизация должна уменьшать возможности к аппроксимации. В следующей секции мы сравним точности спектральных нейронных операторов с нейронными операторами без дискретизации.

4. ЧИСЛЕННОЕ СРАВНЕНИЕ ТОЧНОСТИ НЕЙРОННЫХ И СПЕКТРАЛЬНЫХ НЕЙРОННЫХ ОПЕРАТОРОВ

В данной секции мы сравним нейронный оператор Фурье (FNO) [13], глубокую операторную сеть DeepONet [10], спектральный нейронный оператор SNO(Ch), определенный (16) в базисе Чебышева, SNO(F), определенный (16) в базисе Фурье. Точное описание архитектур и код можно найти в репозитории https://github.com/VLSF/SNOhttps://github.com/VLSF/SNO.

В качестве случайных функций мы используем

$$R(k_{\min}, k_{\max}, \sigma) = \left\{ \frac{\mathcal{R}\left(\sum_{k=k_{\min}}^{k_{\max}} d_k e^{i\pi kx}\right)}{\sqrt{\sum_{k=k_{\min}}^{k_{\max}} \left|d_k^j\right|^2}} \right\}, \quad (17)$$
$$\mathcal{R}(d), \mathcal{I}(d^j) \sim \mathcal{N}(0, \sigma^2),$$

где $\mathcal{R}(\cdot)$, $\mathcal{I}(\cdot)$ обозначает действительную и мнимую части, $\mathcal{N}(a,b)$ – нормальное распределение со средним a и дисперсией b. Операторы, которые были использованы для тестов, описаны ниже.

4.1. Операторы

4.1.1. Интегрирование, сдвиг и дифференцирование. Первые три оператора

$$f(x) \to \int f(y)dy, \quad f(y) \in R(1,10,2);$$
$$f(x) \to f(x)f(x+1), \quad f(x) \in R(0,15,2);$$
$$f(x) \to \frac{d}{dx} f(x), \quad (18)$$
$$f(x) \in R(0,10,k_{\max}), \quad k_{\max} \in \{10,20\}.$$

Таблица 1. Относительная L_2 ошибка для различных задач. SNO(Ch) – спектральный нейронный оператор в базисе Чебышева, SNO(F) – спектральный нейронный оператор в базисе Фурье (в случае задач $2D$, обозначенных xt, базис Чебышева используется по оси t, если задача не является периодической), FNO [13] – нейронный оператор Фурье, DeepONet [10] – глубокая операторная сеть

Датасет	SNO(Ch)	SNO(F)	FNO	DeepONet
Интегрирование	3.5×10^{-4}	$\mathbf{2.6 \times 10^{-4}}$	3.3×10^{-2}	4.7×10^{-2}
∂_x, $k_{\max} = 10$	$\mathbf{1.5 \times 10^{-4}}$	1.3×10^{-3}	5.0×10^{-4}	2.5×10^{-2}
∂_x, $k_{\max} = 20$	3.5×10^{-4}	$\mathbf{2.6 \times 10^{-4}}$	3.3×10^{-2}	4.7×10^{-2}
$f \to f(x)f(x+1)$	$\mathbf{8.5 \times 10^{-3}}$	6.5×10^{-2}	4.6×10^{-2}	1.8
параметрическое ОДУ (DETEST)	$\mathbf{9.7 \times 10^{-4}}$	2.2×10^{-3}	1.1×10^{-3}	2.3×10^{-2}
Эллиптическое уравнение (20)	$\mathbf{1.5 \times 10^{-2}}$	–	2.3×10^{-2}	2.2×10^{-1}
Бюргерс$_x$, $\nu = 0.01$	5.8×10^{-1}	3.1×10^{-1}	$\mathbf{6.3 \times 10^{-2}}$	6.0×10^{-1}
Бюргерс$_x$, $\nu = 0.1$	$\mathbf{1.1 \times 10^{-2}}$	2.8×10^{-2}	1.5×10^{-2}	1.4×10^{-1}
КдВ$_x$ (23)	3.2×10^{-2}	2.3×10^{-2}	$\mathbf{1.3 \times 10^{-3}}$	4.6×10^{-1}
КдВ$_x$ (24)	1.0×10^{-1}	$\mathbf{8.0 \times 10^{-2}}$	9.0×10^{-2}	2.3×10^{-1}
Эллиптическое уравнение (21)	3.0×10^{-2}	–	$\mathbf{2.7 \times 10^{-2}}$	9.3×10^{-2}
Бюргерс$_{xt}$, $\nu = 0.01$	1.9×10^{-1}	2.6×10^{-1}	$\mathbf{4.1 \times 10^{-2}}$	5.8×10^{-1}
Бюргерс$_{xt}$, $\nu = 0.1$	3.4×10^{-2}	9.4×10^{-2}	$\mathbf{6.5 \times 10^{-3}}$	4.1×10^{-1}
КдВ$_{xt}$ (23)	$\mathbf{5.7 \times 10^{-3}}$	5.8×10^{-3}	6.5×10^{-3}	7.1×10^{-2}
КдВ$_{xt}$ (24)	$\mathbf{1.9 \times 10^{-2}}$	2.1×10^{-2}	6.2×10^{-2}	1.6×10^{-1}
Нелинейное уравнение Шрёдингера (25)	$\mathbf{1.4 \times 10^{-2}}$	5.2×10^{-2}	2.6×10^{-2}	1.5×10^{-1}

4.1.2. Параметрическое Обыкновенное Дифференциальное Уравнение. В данном случае мы ищем решение параметрического уравнения $f(t) \to y(t)$ (модификация задачи $A4$ из списка DETEST [1])

$$\frac{dy(t)}{dt} = y(t)f(t), \quad y(0) = \exp\left(\int^{t=0} f(\tau)d\tau\right), \quad (19)$$
$$f(t) \in R(1,30,2).$$

4.1.3. Эллиптическое уравнение. Мы рассматриваем два эллиптических уравнения. В каждом случае задача предсказать решения по коэффициенту диффузии $k(x)$. В $1D$ мы используем уравнение

$$-\frac{d}{dx}k(x)\frac{d}{dx}u(x) = 1, \quad u(-1) = u(1) = 0, \quad (20)$$
$$k(x) = 10(\tanh(g(x)) + 1) + 1, \quad g(x) \in R(0,20,2).$$

В $2D$ мы рассматриваем

$$-\left(\frac{d}{dx}k(x)k(y)\frac{d}{dx} + \frac{d}{dy}k(x)k(y)\frac{d}{dy}\right)u(x,y) = 1, \quad (21)$$
$$k(x) = \left(3(\tanh(g(x)) + 1) + 1\right),$$

внутри $[-1, 1]^2$ с однородными граничными условиями Дирихле и $g(x)$, $g(y)$ из $R(0,20,2)$.

4.1.4. Уравнение Бюргерса. В данном случае задача состоит в поиске решения уравнения $\frac{\partial}{\partial t}u(x,t) + \frac{1}{2}\frac{\partial}{\partial x}u(x,t)^2 - \nu\frac{\partial^2}{\partial x^2}u(x,t) = 0$, $u(x, 0) = f(x)$, $x \in [-1,1]$, $f(x) \in R(0,20,2)$. В $1D$ мы приближаем отображение $f(x) \to u(x, t = 1)$, в $2D$ – отображение $f(x) \to u(x,t)$, $t \in [0,1]$, $x \in [-1,1]$. Мы рассматриваем случай с двумя коэффициентами вязкости $\nu = 0.01$ и $\nu = 0.1$.

4.1.5. Кортевега–де Фриза. Мы рассматриваем уравнение КдВ

$$\frac{\partial}{\partial t}u(x,t) + p\frac{\partial}{\partial x}u(x,t)^2 + \frac{\partial^3}{\partial x^3}u(x,t), \quad (22)$$
$$u(x,0) = f(x).$$

Более точно, используются два решения:

1. Распространение солитона ($p = 1$) [2]:

$$u(x,t) = 3\left(\frac{a}{\cosh(a(x + x_0)/2 - a^3t/2)}\right)^2, \quad (23)$$

где a определяет амплитуду и скорость перемещения, x_0 определяет положение максимума при $t = 0$. Амплитуда a следует однородному распределению [10, 25], f x_0 – однородному распределению на $[-1, 1]$. Цель состоит в предсказании $u(x, t = 0.001)$ из $u(x, t = 0)$ в $1D$ случае, и $u(x, t)$, $t \in [0, 0.001]$ из $u(x, t = 0)$ в $2D$ случае.

2. Взаимодействие двух солитонов ($p = 6$) [3]:

$$\phi_i = a_i(x + x_0^i) - 4a_i^3 t,$$
$$\psi_1(x,t) = 2a_2^2(a_1^2 - a_2^2)\sinh\phi_1,$$
$$\psi_2(x,t) = 2a_1^2(a_1^2 - a_2^2)\cosh\phi_2, \qquad (24)$$
$$u(x,t) = \frac{\psi_1(x,t) + \psi_2(x,t)}{\left(a_1\cosh\phi_1\cosh\phi_2 - a_2\sinh\phi_1\sinh\phi_2\right)^2},$$

здесь a_i определяет скорости, x_0^i начальные положения. Амплитуда a_1 следует однородному распределению на [10, 25], $a_2 = \chi a_1$, где χ следует однородному распределению [1/2, 1], начальные положения фиксированы $x_0^1 = -0.6$, $x_0^2 = -0.5$. Задача состоит в предсказании решения в момент времени $t = 0.005$ или на всем промежутке $t \in [0, 0.005]$, $x \in [-1,1]$.

4.1.6. Нелинейное уравнение Шрёдингера. Мы рассматривает солитон Кузнецова-Ма [8], который имеет вид

$$\psi(x,t) = \left(\frac{-p^2\cos(\omega t) - 2ip\nu\sin(\omega t)}{2\cos(\omega t) - 2\nu\cosh(px/\sqrt{2})} - 1\right)e^{it}, \qquad (25)$$
$$p = 2\sqrt{\nu^2 - 1}, \quad \omega = p\nu.$$

Цель состоит в том, чтобы предсказать $\|\psi(x,t)\|$ on $x \in [-1,1]$, $t \in [0,5]$ из начальных условий. Единственный параметр ν следует однородному распределению [1.5, 3.5].

4.2. Обсуждение результатов

Результаты можно найти в табл. 1. В целом видно, что за исключением уравнения Бюргерса с $\nu = 0.01$ спектральный нейронный оператор либо не сильно уступает нейронному оператору Фурье, либо превосходит его в точности. Таким образом, оказывается, что дискретизация не оказывает существенного влияния на точность нейронного оператора для рассмотренных задач.

5. ЗАКЛЮЧЕНИЕ

В статье были предоставлены аргументы в пользу того, что нейронные операторы на самом деле осуществляют отображения между конечномерными пространствами, неявно определенными выбранной архитектурой и данными, использованными при обучении. Продемонстрировано, что явная дискретизация области определения и значения оператора не приводит к существенному падению точности.

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Исследование выполнено за счет гранта Министерство науки и высшего образования Российской Федерации (проект № 075-10-2021-068).

СПИСОК ЛИТЕРАТУРЫ

1. *Hull T.E. et al.* “Comparing numerical methods for ordinary differential equations”. B: SIAM Journal on Numerical Analysis 9.4. 1972. P. 603–637.
2. *Fornberg B., Whitham G.B.* “A numerical and theoretical study of certain nonlinear wave phenomena”. B: Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences 289.1361. 1978. P. 373–404.
3. *Tohru Yoneyama*. “The Korteweg-de Vries two-soliton solution as interacting two single solitons”. B: Progress of theoretical physics 71.4. 1984. P. 843–846.
4. *Lagaris I.E., Likas A., Fotiadis D.I.* “Artificial neural networks for solving ordinary and partial differential equations”. B: IEEE transactions on neural networks 9.5. 1998. P. 987–1000.
5. *Trefethen L.N.* Spectral methods in MATLAB. SIAM, 2000.
6. *Boyd J.P.* Chebyshev and Fourier spectral methods. Courier Corporation, 2001.
7. *Saksman M.L., Siltanen S. et al.* “Discretization-invariant Bayesian inversion and Besov space priors”. B: arXiv preprint arXiv:0901.4220. 2009.
8. *Kedziora D.J., Ankiewicz A., Akhmediev N.* “Second-order nonlinear Schrödinger equation breather solutions in the degenerate and rogue wave limits”. B: Physical Review E 85.6. 2012. P. 066601.
9. *Bar-Sinai Y. et al.* “Learning data-driven discretizations for partial differential equations”. B: Proceedings of the National Academy of Sciences 116.31. 2019. P. 15344–15349.
10. *Lu Lu, Jin P., Karniadakis G.E.* “DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators”. B: arXiv preprint arXiv:1910.03193. 2019.
11. *Trefethen L.N.* Approximation Theory and Approximation Practice, Extended Edition. SIAM, 2019.
12. *Ummenhofer B. et al.* “Lagrangian fluid simulation with continuous convolutions”. B: International Conference on Learning Representations. 2019.
13. *Zongyi Li et al.* “Fourier neural operator for parametric partial differential equations”. B: arXiv preprint arXiv:2010.08895 (2020).
14. *Zongyi Li et al.* “Neural operator: Graph kernel network for partial differential equations”. B: arXiv preprint arXiv:2003.03485 (2020).
15. *Gupta G., Xiongye Xiao, Bogdan P.* “Multiwavelet-based operator learning for differential equations”. B: Advances in Neural Information Processing Systems. 2021. V. 34. P. 24048–24062.
16. *Kovachki N. et al.* “Neural operator: Learning maps between function spaces”. B: arXiv preprint arXiv:2108.08481 (2021).
17. *Tripura T., Chakraborty S.* “Wavelet neural operator: a neural operator for parametric partial differential equations”. B: arXiv preprint arXiv:2205.02191. 2022.

SPECTRAL NEURAL OPERATORS

V. S. Fanaskov[a] and I. V. Oseledets[a,b]

[a]*Skoltech, Moscow, Russia*

[b]*AIRI, Moscow, Russia*

Presented by Academician of the RAS A.A. Shananin

In recent works, authors introduced neural operator – a particular type of neural network that can approximate maps between infinite-dimensional spaces. Using numerical and analytical techniques, we will highlight the peculiarities of the training and evaluation of these operators. In particular, we will show that for a broad class of neural operators based on integral transformations, a systematic bias is inevitable, owning to aliasing errors. To avoid this bias, we introduce *spectral neural operators* based on explicit discretization of domain and codomain. Discretization should decrease the approximation properties, but numerical experiments show that the accuracy of spectral neural operators is often superior to the one of neural operators defined on infinite-dimensional Banach spaces.

Keywords: neural operators, PDEs