

УДК 517.9

ОБ ОСОБЕННОСТЯХ ЧИСЛЕННОГО ПОДХОДА ПОСТРОЕННОГО НА НЕЙРОННЫХ СЕТЯХ С ПРЯМОЙ СВЯЗЬЮ ДЛЯ РЕШЕНИЯ ЗАДАЧ ДЛЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

© 2024 г. С. А. Ладыгин^а, *, Р. Н. Карачурин^а, П. Н. Рябов^а, Н. А. Кудряшов^а

^аНациональный исследовательский ядерный университет “МИФИ”, Москва, 115409 Россия

*E-mail: SALadygin@mephi.ru

Поступила в редакцию 12.05.2023 г.

После доработки 14.05.2023 г.

Принята к публикации 18.05.2023 г.

На сегодняшний день разработано множество методов численного решения задач, в основе которых лежат обыкновенные дифференциальные уравнения (ОДУ) и уравнения в частных производных (УЧП). Самые распространенные из них это конечно-разностный метод, метод конечных элементов и метод конечных объемов. В данной работе реализован альтернативный численный подход, базирующийся на аппроксимации функций нейронными сетями с прямой связью. Полученное с использованием такого подхода решение, представляет собой дифференцируемое аналитическое выражение чем существенно отличается от других методов, предлагающих либо дискретное решение, либо решение с ограниченной дифференцируемостью. В работе проведено исследование влияния параметров нейронной сети (таких, как функции активации и веса в функции ошибок) на скорость сходимости и точность полученной аппроксимации решения для трех типов дифференциальных уравнений: обыкновенные дифференциальные уравнения, интегрируемые дифференциальные уравнения в частных производных и неинтегрируемые дифференциальные уравнения в частных производных. В качестве модельных уравнений в работе рассматривались уравнения в частных производных Кортевега–де Вриза и Кудряшова–Синельщикова, а также обыкновенное дифференциальное уравнение второго порядка. В каждом вышеописанном случае найдены оптимальные соотношения между весовыми коэффициентами. Установлены наиболее эффективные функции активации для каждой задачи.

Ключевые слова: дифференциальные уравнения, аппроксимация, нейронные сети, численные методы, функции активации

DOI: 10.56304/S2079562923030211

ВВЕДЕНИЕ

При описании многих процессов в физике возникает потребность в решении различных задач в основе модели которых, используются дифференциальные уравнения. Например: одномерная нестационарная задача магнитной гидродинамики [1], упругопластическая задача [2], двумерная задача переноса теплового излучения в анизотропно рассеивающей среде [3], задача конвекции-диффузии [4], задача по моделированию процессов локализации полос адиабатического сдвига в материалах при деформациях [5]. К настоящему времени разработано множество методов численного решения дифференциальных уравнений. Одни выдают решение в виде массива, который содержит значения решения в определенной группе точек (конечно-разностный метод [6]). Другие используют базисные функции для представления решения в аналитической форме и преобразования исходной задачи обычно в систе-

му линейных уравнений (метод конечных элементов [7]). Третьи основываются на интегрировании исходного уравнения по контрольному объему – части расчетной области, примыкающей к заданному расчетному узлу (метод конечных объемов [8, 9]).

В данной работе рассматривается один из методов решения задач как для обыкновенных дифференциальных уравнений (ОДУ), так и уравнений в частных производных (УЧП), который опирается на возможности аппроксимации функций нейронными сетями с прямой связью. Эта идея была впервые описана Лагарисом и др. [10]. Одну из реализаций данной идеи можно найти в работе [11].

Подход предложенный Лагарисом и соавторами заключается в использовании нейронной сети с прямой связью в качестве базового элемента аппроксимации, параметры которой (веса и смещения) настраиваются таким образом, чтобы минимизировать соответствующую функцию ошибки.

При использовании данного метода возникает ряд преимуществ:

1. Решение представляет собой дифференцируемое аналитическое выражение, легко используемое в любых последующих вычислениях. Большинство других методов предлагают дискретное решение (например, методы предиктора–корректора или Рунге–Кутты) или решение с ограниченной дифференцируемостью (например, метод конечных элементов).

2. Требуемое количество параметров модели намного меньше, чем при любом другом методе решения, и поэтому получаются компактные модели решений, которые занимают гораздо меньше места в оперативной памяти компьютера.

3. Метод является общим и может быть применен к ОДУ, а также к УЧП.

4. Этот метод может быть эффективно реализован, используя распараллеливание ресурсов, а также выполняться на графических ускорителях.

Данная работа организована следующим образом. В первом разделе мы приводим описание используемого метода. Второй и третий разделы посвящены применению данного метода к ОДУ и УЧП соответственно, сравнению полученных результатов с точным аналитическим решением, а также анализу эффективности различных нейронных сетей при использовании в них разных функций активаций и влияния весов в функции ошибок.

1. ОПИСАНИЕ МЕТОДА

Рассмотрим следующее дифференциальное уравнение в общем виде:

$$G(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}), \nabla^2 u(\mathbf{x}), \dots, \nabla^m u(\mathbf{x})) = 0, \quad \mathbf{x} \in D, \quad (1)$$

где $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, $D \subset \mathbb{R}^n$ – область определения, а $u(\vec{x})$ – искомое решение. Для уравнения (1) зададим граничные условия в виде:

$$B(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}), \nabla^2 u(\mathbf{x}), \dots, \nabla^{m-1} u(\mathbf{x})) = 0, \quad \mathbf{x} \in \partial D. \quad (2)$$

Нейросеть будет приближать решение к истинному в каком-то заданном наборе точек в области D и на ее границе ∂D . Для этого из области D и ее границы ∂D выделим дискретные множества точек \hat{D} и $\widehat{\partial D}$. Введем функцию $y(\mathbf{x})$, которая считается с помощью обучаемой нейронной сети и является аппроксимацией функции $u(\mathbf{x})$. Тогда при обучении нейронная сеть должна стремиться к тому, чтобы выполнялись следующие выражения:

$$\begin{cases} G(\mathbf{x}_i, y(\mathbf{x}_i), \nabla y(\mathbf{x}_i), \nabla^2 y(\mathbf{x}_i), \dots, \nabla^m y(\mathbf{x}_i)) = 0, \\ \forall \mathbf{x}_i \in \hat{D}, \\ B(\mathbf{x}_j, y(\mathbf{x}_j), \nabla y(\mathbf{x}_j), \nabla^2 y(\mathbf{x}_j), \dots, \nabla^{m-1} y(\mathbf{x}_j)) = 0, \\ \forall \mathbf{x}_j \in \widehat{\partial D}. \end{cases} \quad (3)$$

Нейронная сеть ищет параметры (такие как веса и смещения) таким образом, чтобы минимизировать функцию ошибок (далее – *loss*), которая имеет следующий вид:

$$loss = \lambda_b \sigma_b^2 + \lambda_e \sigma_e^2, \quad (4)$$

где

$$\sigma_b^2 = \sum_{\mathbf{x}_j \in \widehat{\partial D}} [B(\mathbf{x}_j, y(\mathbf{x}_j), \nabla y(\mathbf{x}_j), \nabla^2 y(\mathbf{x}_j), \dots, \nabla^{m-1} y(\mathbf{x}_j))]^2, \quad (5)$$

$$\begin{aligned} \sigma_e^2 = \\ = \sum_{\mathbf{x}_i \in \hat{D}} [G(\mathbf{x}_i, y(\mathbf{x}_i), \nabla y(\mathbf{x}_i), \nabla^2 y(\mathbf{x}_i), \dots, \nabla^m y(\mathbf{x}_i))]^2. \end{aligned} \quad (6)$$

Здесь σ_e^2 представляет собой среднеквадратичную ошибку на внутренних точках, а σ_b^2 среднеквадратичную ошибку на границе, λ_b и λ_e представляют собой веса, которые определяют приоритеты приближению значений функции на границе или же внутри области. Чаще всего их значение выбирают равным единице, но в данной работе мы рассмотрим влияние этих коэффициентов на точность и скорость сходимости полученных результатов.

Как видно из уравнений (5) и (6), для того, чтобы получить значение функции *loss* необходимо уметь искать градиенты от функции $y(\mathbf{x})$. Есть несколько подходов позволяющих сделать это, но самый эффективный и универсальный – это автотифференцирование [12]. Пусть нам известны некие “базисные” функции и значения их производных. Тогда для любой сложной функции, которая определяется через эти “базисные” функции, мы можем автоматически вычислить ее значение и значение ее производных. Для нашего случая функция $y(\mathbf{x})$, которая аппроксимирует решение $u(\mathbf{x})$, является суперпозицией функций активации, которые и служат теми “базисными” функциями. Поэтому отсюда следует естественное ограничение на эти функции.

Замечание Для аппроксимации решения $u(\mathbf{x})$ уравнения (1) с граничными условиями (2) нейронной сетью, ее функции активации должны быть m раз непрерывно дифференцируемы и их производные не должны быть тождественно равны нулю.

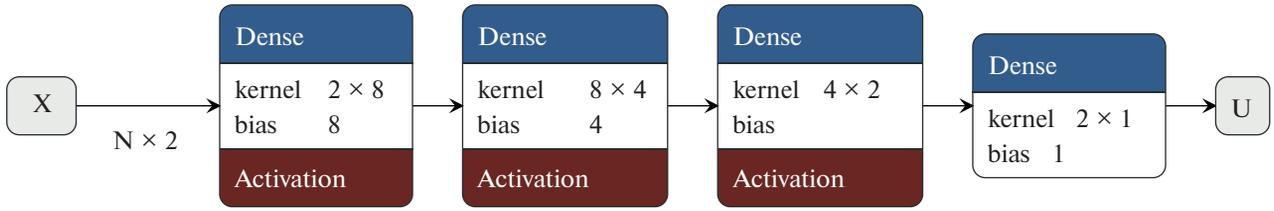


Рис. 1. Модель нейронной сети.

Если это выполнено, тогда все градиенты функции $y(x)$ будут найдены автоматически из известных значений производных функций активации.

2. ЧИСЛЕННОЕ РЕШЕНИЕ ОДНОЙ ЗАДАЧИ ДЛЯ ОБЫКНОВЕННОГО ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ

2.1. Постановка задачи

Рассмотрим частный случай дифференциального уравнения (1), а именно обыкновенное дифференциальное уравнение:

$$G(x, u(x), u'(x), u''(x), \dots, u^{(m)}(x)) = 0, \quad a < x < b. \quad (7)$$

Здесь $u(x)$ – функция вещественной переменной x и искомое решение. Для уравнения (7) зададим граничные условия:

$$\begin{cases} u(a) = u_a \in \text{const}, \\ u(b) = u_b \in \text{const}. \end{cases} \quad (8)$$

Введем равномерную пространственную сетку с N точками и шагом $h = (b - a)/(N - 1)$:

$$\Omega = \left\{ \begin{array}{l} x_i = a + ih; i = 0, 1, \dots, N - 1; \\ x_0 = a; x_{N-1} = b \end{array} \right\}. \quad (9)$$

А также функцию $y(x)$, которая является аппроксимацией функции $u(x)$ и считается с помощью обучаемой нейронной сети. Тогда в процессе обучения нейронная сеть будет стремиться удовлетворить следующие выражения:

$$\begin{cases} G(x_i, y(x_i), y'(x_i), y''(x_i), \dots, y^{(m)}(x_i)) = 0, \\ \forall x_i \in \Omega, \\ y(a) = u_a, \\ y(b) = u_b. \end{cases} \quad (10)$$

Отсюда в функции $loss$ (4) среднеквадратичные ошибки σ_a^2 и σ_b^2 будут иметь вид:

$$\begin{cases} \sigma_e^2 = \sum_{x_i \in \Omega} [G(x_i, y(x_i), y'(x_i), y''(x_i), \dots, y^{(m)}(x_i))]^2, \\ \sigma_b^2 = (y(a) - u_a)^2 + (y(b) - u_b)^2. \end{cases} \quad (11)$$

2.2. Пример

Рассмотрим следующую краевую задачу:

$$\begin{cases} u''(x) + \sin(x) = 0, & 0 < x < \pi, \\ u(0) = 0, \\ u(\pi) = 0. \end{cases}$$

Аналитическое решение данной задачи имеет вид:

$$u(x) = \sin(x). \quad (12)$$

Рассмотрим пространственную сетку (9) со 100 точками и шагом $h = \pi/99$:

$$\Omega = \{x_i = ih; i = 0, 1, \dots, 99; x_0 = 0; x_{99} = \pi\}.$$

Тогда функция $loss$, согласно (11), имеет следующий вид:

$$loss = \lambda_e \sum_{x_i \in \Omega} [y''(x_i) + \sin(x_i)]^2 + \lambda_b (y(0)^2 + y(\pi)^2).$$

В качестве оптимизатора нейронной сети был использован метод Adam. Модель данной нейронной сети представлена на рис. 1.

Для определения оптимальной модели нейронной сети, было проведено несколько обучений с различными функциями активаций. Значения весов λ_e и λ_b были взяты за единицу. Результаты обучений можно увидеть на графике функции $loss$ (рис. 2).

Лучшие результаты показали функции активации \sin и \tanh .

Функция активации	Значение loss
\sin	$4.1846524 \cdot 10^{-5}$
\tanh	0.0001570559

Для нейронных сетей с данными функциями активаций были проведены дополнительные обучения с разными отношениями весов λ_e и λ_b . Были рассмотрены отношения:

$$\frac{\lambda_e}{\lambda_b} = 2^{-4}, 2^{-3}, \dots, 2^3, 2^4.$$

В результате были получены графики (см. рис. 3) зависимости значения невязки от отношения весов.

Лучший результат, который удалось получить, в процессе обучения показала модель с функцией

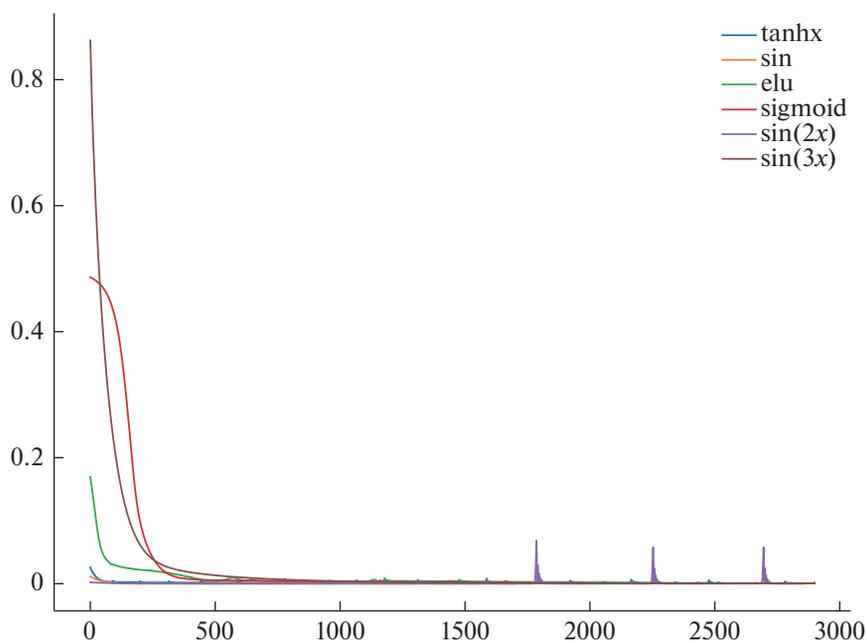


Рис. 2. График функции *loss* для различных функций активаций при числе эпох 3000.

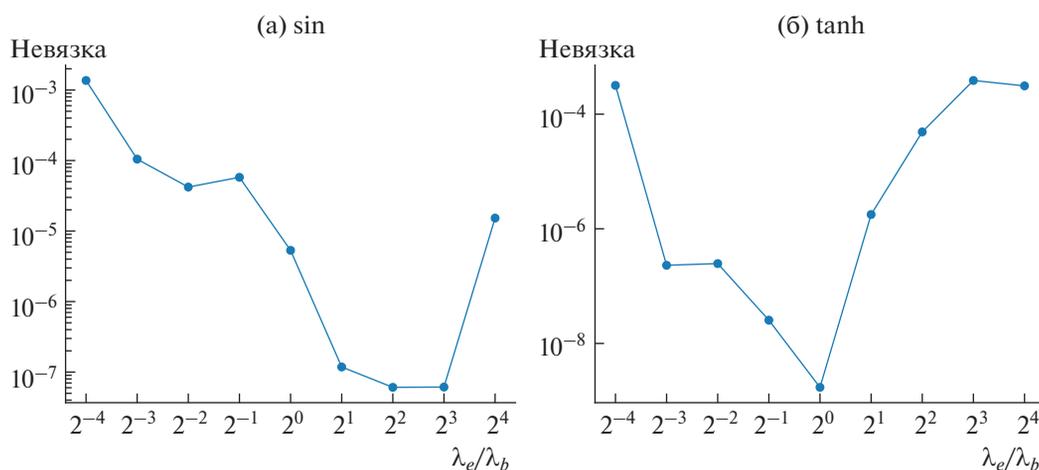


Рис. 3. Графики зависимости значения невязки от отношения весов.

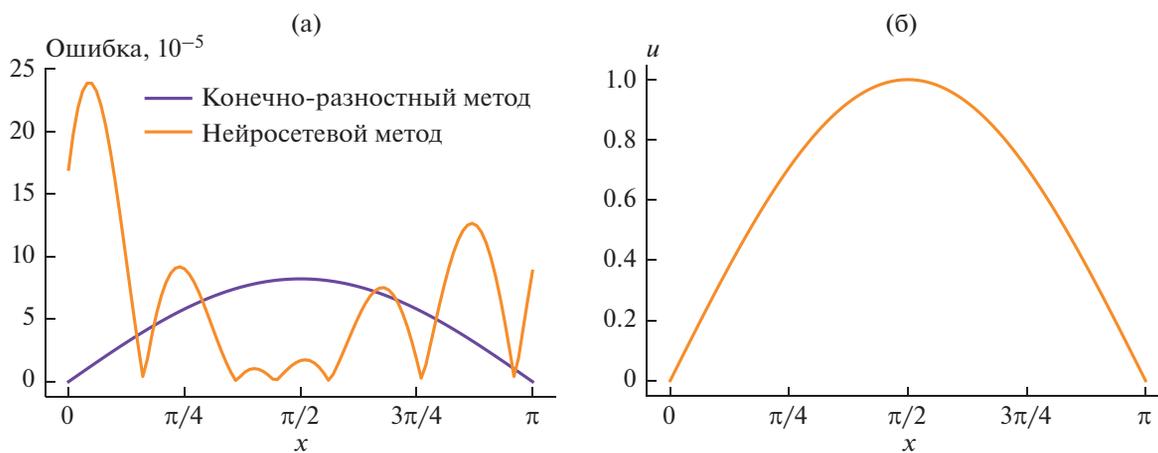


Рис. 4. График абсолютной ошибки (а), график решения (б).

активации \tanh и отношением $\frac{\lambda_e}{\lambda_b} = 2^0$. Для наглядности построим график абсолютной ошибки для аппроксимации с использованием данной модели (см. рис. 4), зная аналитическое решение (12). А также сравним с абсолютной ошибкой при использовании конечно-разностного метода.

Из приведенного графика видно, что по порядку точности нейросетевой метод и метод конечных разностей дает схожие результаты.

3. ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧ ДЛЯ УРАВНЕНИЯ КОРТЕВЕГА–ДЕ ВРИЗА И КУДРЯШОВА–СИНЕЛЬЩИКОВА

3.1. Постановка задачи

Рассмотрим еще один часто встречающийся частный случай дифференциального уравнения (1), а именно дифференциальное уравнение в частных производных, в котором решение $u(x, t)$ является функцией времени и координаты:

$$G\left(x, t, u(x, t), \frac{\partial u(x, t)}{\partial t}, \frac{\partial u(x, t)}{\partial x}, \dots, \frac{\partial^m u(x, t)}{\partial x^m}\right) = 0, \quad (13)$$

$t > 0, \quad a < x < b.$

Здесь $u(x, t)$ – функция вещественных переменных x и t , а также искомое решение. Для уравнения (13) зададим граничные условия:

$$\begin{cases} u(x, 0) = u_0(x) \in C^0[a, b], \\ B_a\left(t, u(a, t), \frac{\partial u(x, t)}{\partial x}\Big|_{x=a}, \dots, \frac{\partial^m u(x, t)}{\partial x^m}\Big|_{x=a}\right) = 0, \\ B_b\left(t, u(b, t), \frac{\partial u(x, t)}{\partial x}\Big|_{x=b}, \dots, \frac{\partial^m u(x, t)}{\partial x^m}\Big|_{x=b}\right) = 0. \end{cases} \quad (14)$$

Введем равномерную сетку по переменной x с шагом $h = (b - a)/(N - 1)$:

$$\omega_h = \{x_i = a + ih; i = 0, 1, \dots, N - 1; x_0 = a; x_{N-1} = b\}, \quad (15)$$

и равномерную сетку по переменной t с шагом $\tau = T/(K - 1)$:

$$\omega_\tau = \{t_n = n\tau; n = 0, 1, \dots, K - 1; t_{K-1} = T\}. \quad (16)$$

Точки $(x_i, t_n), i = 0, 1, \dots, N - 1, n = 0, 1, \dots, K - 1$, образуют узлы пространственно-временной сетки:

$$\Omega = \omega_h \times \omega_\tau. \quad (17)$$

Также введем функцию $y(x, t)$, которая является аппроксимацией функции $u(x, t)$ и считается с помощью обучаемой нейронной сети. Тогда в процессе обучения нейронная сеть будет стремиться удовлетворить следующие выражения:

$$\begin{cases} G\left(x_i, t_n, y(x_i, t_n), \frac{\partial y(x, t)}{\partial t}\Big|_{x_i, t_n}, \frac{\partial y(x, t)}{\partial x}\Big|_{x_i, t_n}, \dots, \frac{\partial^m y(x, t)}{\partial x^m}\Big|_{x_i, t_n}\right) = 0, \\ y(x_i, 0) = u_0(x_i), \\ B_a\left(t_n, y(a, t_n), \frac{\partial y(x, t)}{\partial x}\Big|_{a, t_n}, \dots, \frac{\partial^m y(x, t)}{\partial x^m}\Big|_{a, t_n}\right) = 0, \\ B_b\left(t_n, y(b, t_n), \frac{\partial y(x, t)}{\partial x}\Big|_{b, t_n}, \dots, \frac{\partial^m y(x, t)}{\partial x^m}\Big|_{b, t_n}\right) = 0, \\ \forall (x_i, t_n) \in \Omega. \end{cases}$$

Отсюда в функции $loss$ (4) среднеквадратичные ошибки σ_e^2 и σ_b^2 будут иметь вид:

$$\begin{cases} \sigma_e^2 = \sum_{(x_i, t_n) \in \Omega} [G(x_i, t_n, y(x_i, t_n), \dots)]^2, \\ \sigma_b^2 = \sum_{x_i \in \omega_h} (y(x_i, 0) - u_0(x_i))^2 + \\ + \sum_{t_n \in \omega_\tau} [B_a(t_n, y(a, t_n), \dots)]^2 + \\ + \sum_{t_n \in \omega_\tau} [B_b(t_n, y(b, t_n), \dots)]^2. \end{cases} \quad (18)$$

3.2. Пример №1 (уравнение Кортевега–де Вриза)

В первом примере рассмотрим следующую краевую задачу для уравнения Кортевега–де Вриза:

$$\begin{cases} u_t + 6uu_x + u_{xxx} = 0, \quad -10 < x < 10, \quad 0 < t < 1, \\ u|_{t=0} = 0.5\text{sech}^2(x/2), \\ u|_{x=-10} = 0.5\text{sech}^2(5 + t/2), \\ u|_{x=10} = 0.5\text{sech}^2(5 - t/2), \\ u_x|_{x=10} = -0.5\tanh(5 - t/2)\text{sech}^2(5 - t/2). \end{cases}$$

Уравнение Кортевега–де Вриза встречается для описания длинных волн на воде. Данное уравнение относится к классу интегрируемых уравнений. Более подробную информацию о данном уравнении можно найти здесь [13]. Выпишем аналитическое решение данной задаче приведенное в [14]:

$$u(x, t) = 0.5\text{sech}^2(x/2 - t/2). \quad (19)$$

Рассмотрим пространственную сетку (17) со 100 точками по переменной x и с 6 точками по переменной t . Тогда функция $loss$, согласно (18), имеет следующий вид:

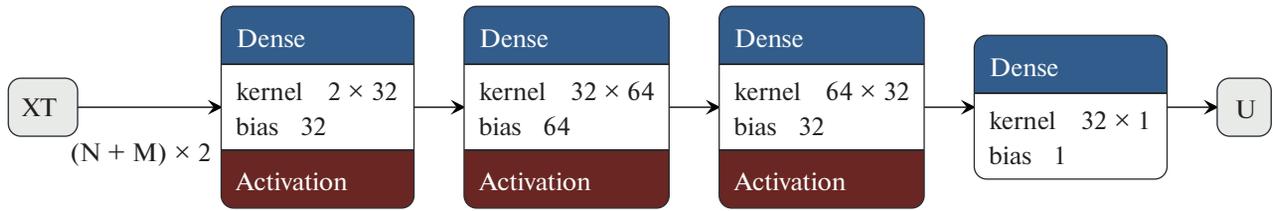


Рис. 5. Модель нейронной сети.

$$\begin{aligned}
 loss = & \\
 = \lambda_e \sum_{x_i, t_n \in \Omega} & \left[y_t|_{(x_i, t_n)} + 6 y_x|_{(x_i, t_n)} y_x|_{(x_i, t_n)} + y_{xxx}|_{(x_i, t_n)} \right]^2 + \\
 & + \lambda_b \sum_{x_i \in \omega_h} \left[y|_{(x_i, 0)} - 0.5 \operatorname{sech}^2(x_i/2) \right]^2 + \\
 & + \lambda_b \sum_{t_n \in \omega_\tau} \left[y|_{(-10, t_n)} - 0.5 \operatorname{sech}^2(5 + t_n/2) \right]^2 + \\
 & + \lambda_b \sum_{t_n \in \omega_\tau} \left[y|_{(10, t_n)} - 0.5 \operatorname{sech}^2(5 - t_n/2) \right]^2 + \\
 & + \lambda_b \sum_{t_n \in \omega_\tau} \left[y_x|_{(10, t_n)} + \right. \\
 & \left. + 0.5 \operatorname{tanh}(5 - t_n/2) \operatorname{sech}^2(5 - t_n/2) \right]^2.
 \end{aligned}$$

В качестве оптимизатора нейронной сети был использован метод Adam. Модель данной нейронной сети представлена на рис. 5.

Для определения оптимальной модели нейронной сети, было проведено несколько обучений с различными функциями активаций. Значения весов λ_e и λ_b были взяты за 1. Результаты обучений можно увидеть на графике функции $loss$ (рис. 6).

Лучший результат показала функция активации \tanh со значением $loss$ за 10000 эпох равным $8.4624990 \cdot 10^{-5}$.

Для нейронной сети с данной функцией активации были проведены дополнительные обучения с разными отношениями весов λ_e и λ_b . Были рассмотрены отношения:

$$\frac{\lambda_e}{\lambda_b} = 2^{-4}, 2^{-3}, \dots, 2^3, 2^4.$$

В результате был получен график (см. рис. 7) зависимости значения невязки на эпохе 10000 от отношения весов.

Отсюда видно, что лучший результат в процессе обучения показала модель с функцией актива-

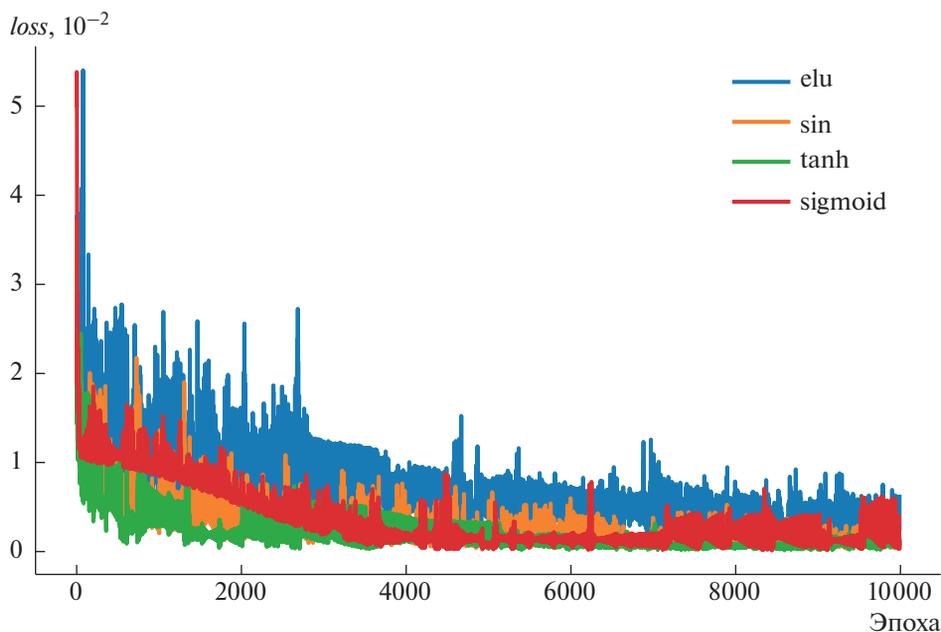


Рис. 6. График функции $loss$ для различных функций активаций при числе эпох 10000.

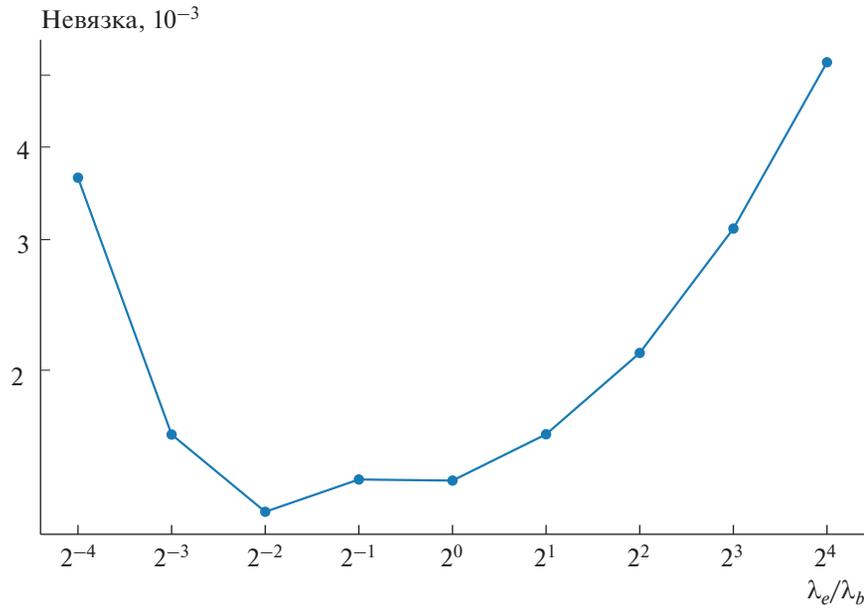


Рис. 7. Графики зависимости значения невязки на эпохе 10000 от отношения весов.

ции \tanh и отношением $\frac{\lambda_e}{\lambda_b} = 2^{-2}$. Для наглядности построим график абсолютной ошибки в последний момент времени для аппроксимации с использованием данной модели (см. рис. 8), зная аналитическое решение (19). А также сравним с абсолютной ошибкой численного метода линий.

3.3. Пример №2 (уравнение Кудряшова–Синельщикова)

Для удобства представления задачи второго примера введем функцию двух переменных, схожую с сигмоидой:

$$\sigma_c(x, t) \equiv \frac{1}{1 + \exp(2x + 8.24t)}.$$

Рассмотрим следующую краевую задачу для уравнения Кудряшова–Синельщикова:

$$\begin{cases} u_t - 96/25uu_x + u_{xxx} - (uu_{xx})_x + \\ + 3u_xu_{xx} - u_{xx} - 0.4(uu_x)_x = 0, \\ -10 < x < 10, \quad 0 < t < 1, \\ u|_{t=0} = 23/16 - 35/8\sigma_c(x, 0) + 175/48\sigma_c^2(x, 0), \\ u|_{x=-10} = 23/16 - 35/8\sigma_c(-10, t) + \\ + 175/48\sigma_c^2(-10, t), \\ u|_{x=10} = 23/16 - 35/8\sigma_c(10, t) + 175/48\sigma_c^2(10, t), \\ u_x|_{x=10} = 35/4\sigma_c(10, t) - 70/3\sigma_c^2(10, t) + \\ + 175/12\sigma_c^3(10, t). \end{cases} \quad (20)$$

Уравнение Кудряшова–Синельщикова впервые было получено в работе [15]. Данное уравнение используется для описания волн в жидкости с пузырьками газа [15] и относится к классу неинтегрируемых дифференциальных уравнений. Аналитическое решение задачи (20), согласно [16], имеет вид:

$$u(x, t) = 23/16 - 35/8\sigma_c(x, t) + 175/48\sigma_c^2(x, t). \quad (21)$$

Рассмотрим пространственную сетку (17) с 200 точками по переменной x и с 10 точками по переменной t . Тогда функция $loss$, согласно (18), имеет следующий вид:

$$\begin{aligned} loss = \lambda_e \sum_{x_i, t_n \in \Omega} [y_i - 96/25u y_x + y_{xxx} - \\ - (u y_{xx})_x + 3y_x y_{xx} - y_{xx} - 0.4(y y_x)_x]^2 \Big|_{(x_i, t_n)} + \\ + \lambda_b \sum_{x_i \in \omega_h} [y|_{t=0} - 23/16 + 35/8\sigma_c(x, 0) - \\ - 175/48\sigma_c^2(x, 0)]^2 \Big|_{x=x_i} + \lambda_b \sum_{t_n \in \omega_\tau} [y|_{x=-10} - 23/16 + \\ + 35/8\sigma_c(-10, t) - 175/48\sigma_c^2(-10, t)]^2 \Big|_{t=t_n} + \\ + \lambda_b \sum_{t_n \in \omega_\tau} [y|_{x=10} - 23/16 + 35/8\sigma_c(10, t) - \\ - 175/48\sigma_c^2(10, t)]^2 \Big|_{t=t_n} + \lambda_b \sum_{t_n \in \omega_\tau} + [y_x|_{x=10} - \\ - 35/4\sigma_c(10, t) + 70/3\sigma_c^2(10, t) - 175/12\sigma_c^3(10, t)]^2 \Big|_{t=t_n}. \end{aligned}$$

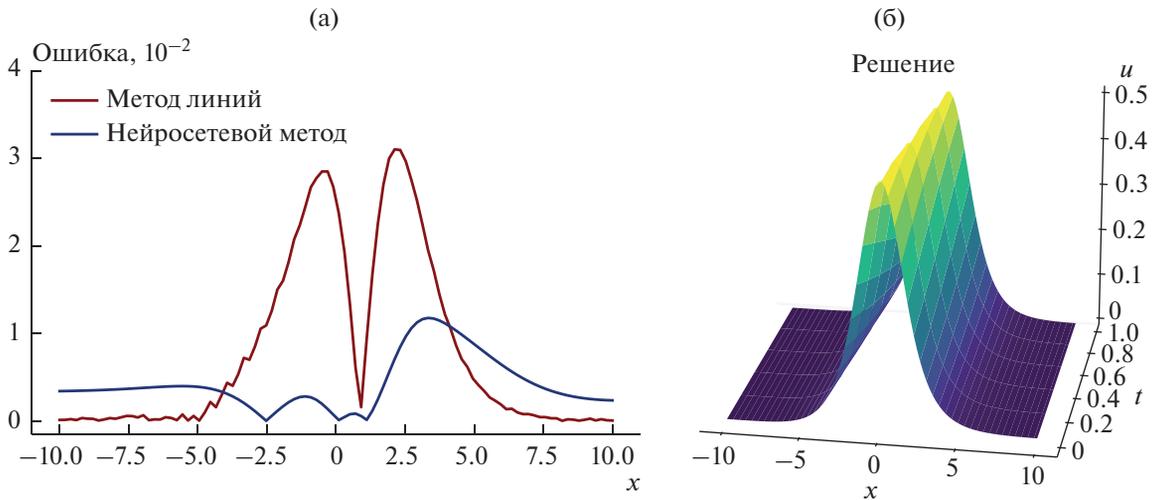


Рис. 8. График абсолютной ошибки в последний момент времени (а), график решения (б).

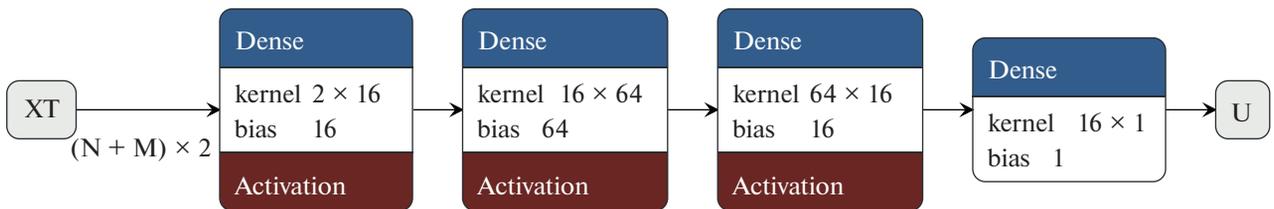


Рис. 9. Модель нейронной сети.

В качестве оптимизатора нейронной сети был использован метод Adam. Модель данной нейронной сети представлена на рис. 9.

Для определения оптимальной модели нейронной сети, было проведено несколько обуче-

ний с различными функциями активаций. Значения весов λ_e и λ_b были взяты за единицу. Результаты обучений можно увидеть на графике функции $loss$ (рис. 10).

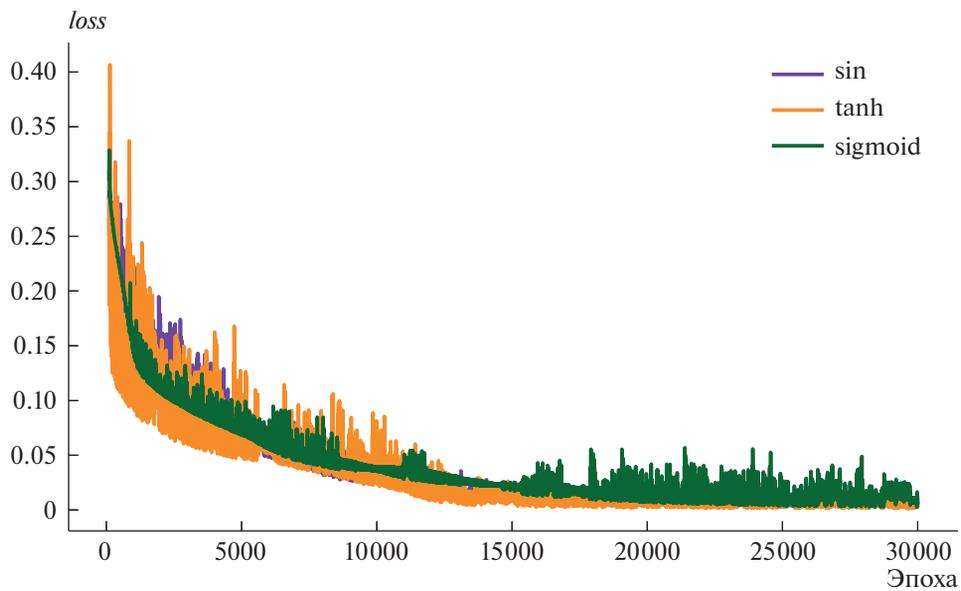


Рис. 10. График функции $loss$ для различных функций активаций при числе эпох 30000.

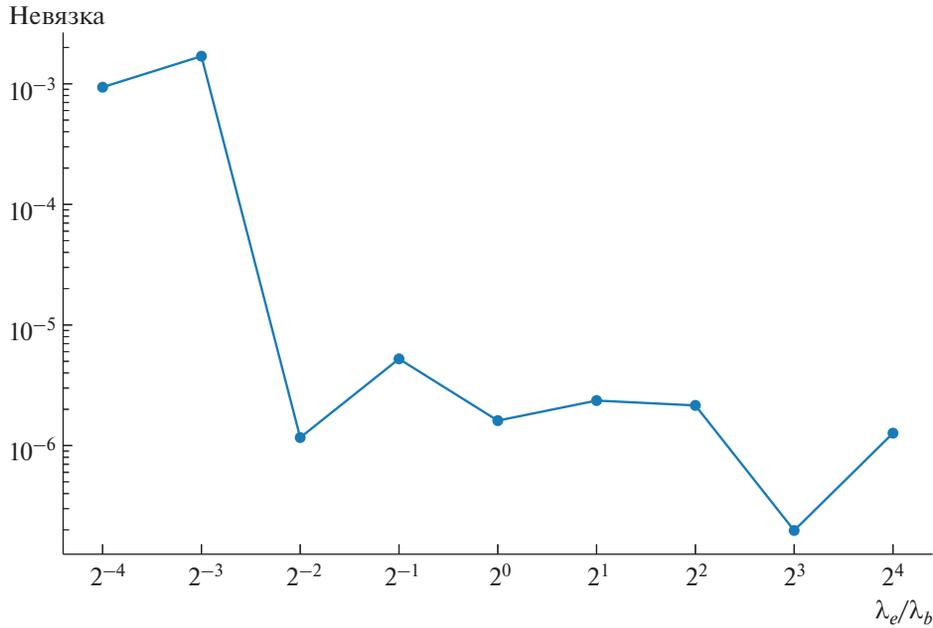


Рис. 11. Графики зависимости значения невязки на эпохе 30000 от отношения весов.

Лучший результат показала функция активации \sin со значением $loss$ за 30000 эпох равным 0.0025825875.

Для нейронной сети с данной функцией активации были проведены дополнительные обучения с разными отношениями весов λ_e и λ_b . Были рассмотрены отношения:

$$\frac{\lambda_e}{\lambda_b} = 2^{-4}, 2^{-3}, \dots, 2^3, 2^4.$$

В результате был получен график (см. рис. 11) зависимости значения невязки на эпохе 30000 от отношения весов.

Отсюда видно, что лучший результат в процессе обучения показала модель с функцией активации \sin и отношением $\frac{\lambda_e}{\lambda_b} = 2^3$. Для наглядности построим график абсолютной ошибки и решения для аппроксимации с использованием данной модели (см. рис. 12), зная аналитическое решение (21).

4. ЗАКЛЮЧЕНИЕ

В данной работе реализован нейросетевой метод, позволяющий строить численные решения задач для обыкновенных дифференциальных уравне-

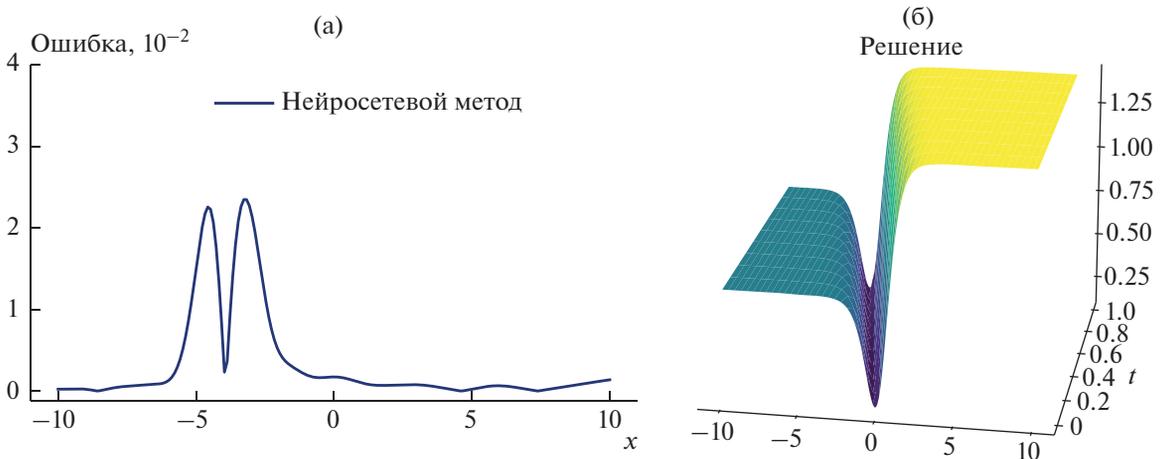


Рис. 12. График ошибки (а), график решения (б).

ний и уравнений в частных производных. В частности, с помощью предложенного метода были решены три модельные задачи: задача о колебаниях математического маятника, задачи для уравнения Кортевега–де Вриза и уравнения Кудряшова–Синельщикова.

В каждом отдельном случае, проведено исследование влияния различных функций активации на полученные результаты. Для рассмотренных задач наиболее эффективно показали себя функции активации в виде синуса $\sin x$ и гиперболического тангенса $\tanh x$.

Также проведено исследование влияния весовых коэффициентов в функции ошибок *loss*. Показано, что оптимальное соотношение данных параметров для каждой рассмотренной задачи свое. Более того, данное соотношение зависит от активационной функции. В частности, для рассмотренных в работе задач, данная величина колебалась в интервале $[2^{-2}, 2^3]$.

БЛАГОДАРНОСТИ

Благодарим Ивлева А.Е. за предоставленные данные по численному решению задачи для уравнения Кортевега–де Вриза методом линий.

Исследование выполнено за счет гранта Российского научного фонда (проект № 21-71-00102).

СПИСОК ЛИТЕРАТУРЫ/REFERENCES

1. Самарский А.А. и др. // Журнал вычислительной математики и математической физики. 1968. Т. 8 (5). С. 1025.
2. Бураго Н.Г., Кукуджанов В.Н. Решение упругопластических задач методом конечных элементов. Пакет прикладных программ “Астра”. 1988. Препринт.
3. Домбровский Л.А., Баркова Л.Г. // Теплофизика высоких температур, 1986. Т. 24 (4). С. 762.
4. Вабищевич П.Н., Самарский А.А. // Журнал вычислительной математики и математической физики. 2000. Т. 40 (5). С. 726.
5. Ryabov P.N. // J. Mech. Phys. Solids. 2015. V. 76. P. 180.
6. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений: учеб. пособие для вузов. 1978. Москва: Наука.
7. Зенкевич О. Метод конечных элементов в технике. 1975. Москва: Мир.
8. Ковеня В.М., Чирко Д.В. Методы конечных разностей и конечных объемов для решения задач математической физики. 2013. Новосибирск: НГУ. С. 24–26.
9. Muratov R.V., Kudryashov N.A., Ryabov P.N. // Commun. Nonlin. Sci. Numer. Simul. 2021. V. 101. P. 105858.
10. Lagaris I.E., Likas A., Fotiadis D.I. // IEEE Trans. Neural Networks. 1998. V. 9 (5). P. 987.
11. Fang Y. et al. // Chaos, Solitons Fract. 2022. V. 158. P. 112118.
12. Ketkar N., Moolayil J. Automatic Differentiation in Deep Learning. Deep Learning with Python. 2021. Berkeley: Apress.
13. Кудряшов Н.А. Методы нелинейной математической физики: Учебное пособие. 2010. Долгопрудный: Издательский Дом “Интеллект”.
14. Li J., Chen Y. // Commun. Theor. Phys. 2020. V. 72 (11). P. 115003.
15. Kudryashov N.A., Sinelshchikov D.I. // Phys. Lett. A. 2010. V. 374. P. 2011.
16. Ryabov P.N. // Appl. Math. Comput. 2010. V. 217 (7). P. 3585.

On Specific Features of an Approach Based on Feedforward Neural Networks to Solve Problems Based on Differential Equations

S. A. Ladygin¹*, R. N. Karachurin¹, P. N. Ryabov¹, and N. A. Kudryashov¹

¹National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, 115409 Russia

*e-mail: SALadygin@mephi.ru

Received May 12, 2023; revised May 14, 2023; accepted May 18, 2023

Abstract—To date, a multitude of methods have been developed for numerical solution of problems based on ordinary differential equations (ODEs) and partial differential equations (PDEs). The most common of these are finite-difference method, the finite-element method, and the finite-volume method. In this study, an alternative numerical approach is implemented, based on the approximation of functions by feedforward neural networks. The solution obtained using this approach is a differentiable analytical expression; in this respect it differs significantly from other methods that offer either discrete solutions or solutions with limited differentiability. In this study, we examine the influence of neural-network parameters (such as activation functions and weights in the error function) on the rate of convergence and accuracy of the obtained approximation of the solution for three types of differential equations: ordinary differential equations, integrable partial differential equations, and non-integrable partial differential equations. As model equations, we consider Korteweg–de Vries and Kudryashov–Sinelshchikov partial differential equations and second-order ordinary differential equations. In each case described above, the optimal ratios of the weight coefficients are found. The activation functions most efficient for each problem are determined.

Keywords: differential equations, approximation, neural networks, numerical methods, activation functions