

Применение методов нейросетевого моделирования при решении начально-краевых задач для дифференциальных уравнений в частных производных

Владимир Евгеньевич Вершинин[✉], Роман Юрьевич Пономарев

Тюменский нефтяной научный центр, Тюмень, Россия
Контакт для переписки: ve_ershinin2@tnnc.rosneft.ru[✉]

Аннотация. Машинное обучение позволяет решать самые различные задачи анализа данных, однако его использование для решения дифференциальных уравнений появилось сравнительно недавно. Аппроксимация решения краевой задачи для дифференциальных уравнений (обыкновенных и в частных производных) при этом строится с помощью нейросетевых функций. Подбор весовых коэффициентов осуществляется в ходе обучения нейронной сети. Критериями качества обучения при этом выступают невязки по уравнению и гранично-начальным условиям. Данный подход позволяет находить вместо сеточных такие решения, которые заданы на всей области допустимых значений краевой задачи. На конкретных примерах показаны особенности применения физико-информированных нейронных сетей к решению краевых задач для дифференциальных уравнений различных типов. Методы обучения физико-информированных нейронных сетей могут быть использованы в задачах дообучения интеллектуальных управляющих систем на неполных наборах входных данных.


Ключевые слова: машинное обучение, физико-информированные нейронные сети, PINN, искусственная нейронная сеть

Цитирование: Вершинин В. Е., Пономарев Р. Ю. 2023. Применение методов нейросетевого моделирования при решении начально-краевых задач для дифференциальных уравнений в частных производных // Вестник Тюменского государственного университета. Физико-математическое моделирование. Нефть, газ, энергетика. Том 9. № 3 (35). С. 132–147. <https://doi.org/10.21684/2411-7978-2023-9-3-132-147>

Поступила 25.09.2023; одобрена 28.09.2023; принята 06.10.2023

Application of neural network modeling methods in solving initial boundary value problems for partial differential equations

Vladimir E. Vershinin , Roman Yu. Ponomarev

Tyumen Petroleum Research Center, Tyumen, Russia
Corresponding author: ve_vershinin2@tnnc.rosneft.ru 

Abstract. Machine learning allows you to solve a variety of data analysis problems, but its use for solving differential equations has appeared relatively recently. The approximation of the solution of the boundary value problem for differential equations (ordinary and partial derivatives) is constructed using neural network functions. The selection of weighting coefficients is carried out during the training of the neural network. The criteria for the quality of training in this case are inconsistencies in the equation and boundary-initial conditions. This approach makes it possible, instead of grid solutions, to find solutions defined on the entire feasible region of the boundary value problem. Specific examples show the features of the application of physics-informed neural networks to the solution of boundary value problems for differential equations of various types. Physics-informed neural networks training methods can be used in the tasks of retraining intelligent control systems on incomplete sets of input data.

Keywords: machine learning, physics-informed neural networks, PINN, artificial neural network

Citation: Vershinin, V. E., & Ponomarev, R. Yu. (2023). Application of neural network modeling methods in solving initial boundary value problems for partial differential equations. *Tyumen State University Herald. Physical and Mathematical Modeling. Oil, Gas, Energy*, 9(3), 132–147. <https://doi.org/10.21684/2411-7978-2023-9-3-132-147>

Received September 25, 2023; Reviewed September 28, 2023; Accepted October 6, 2023

Введение

Широко распространенным подходом при решении краевых задач для дифференциальных уравнений (ДУ) на сегодняшний момент является использование методов конечных разностей. При этом приближенное решение ищется в виде сеточной функции, заданной в узлах сетки. Измельчение шага сетки позволяет достаточно точно смоделировать физические процессы при решении большинства инженерных задач. Однако для этого требуются большие вычислительные мощности и значительное количество машинного

времени. Точность моделирования физических процессов и сходимость численных методов очень сильно зависит от качества построения сеточной области расчета, которая, в свою очередь, существенно зависит от формы области расчета.

Одним из альтернативных методов решения краевых задач и моделирования физических процессов является нейросетевое моделирование. Нейронная сеть в данном случае выполняет роль функции, преобразующей входной вектор в выходной. Обладая высокой гибкостью при параметрической настройке, искусственная нейронная сеть (ИНС) способна выступать в качестве универсального аппроксиматора искомых функций-решений [Хайкин, 2019].

Проблема аппроксимации непрерывных функций с помощью небольшого числа простейших базисных функций имеет давнюю историю. Принципиальная возможность нейросетевой аппроксимации базируется на теореме Арнольда и Колмогорова, в которой доказано, что любую непрерывную функцию переменных n можно получить с помощью операций сложения, умножения и суперпозиции из непрерывных функций одной переменной [Колмогоров, 1956]. Исследовался вопрос о классах функций, аппроксимируемых с любой степенью точности нейронными сетями разных типов [Тархов, 2014; Васильев и др., 2015; Хайкин, 2019]. В частности, на основании теоремы Стоуна показана пригодность сетей радиально-базисных функций (radial basis function networks, RBF networks) и сетей персептронного типа для решения задач аппроксимации. По сравнению с традиционными методами на основе сеток, машинное обучение представляет собой бессеточный подход к аппроксимации искомого решения. Благодаря гибкости и универсальности своего алгоритма, нейросетевое моделирование вызывает повышенный интерес исследователей при рассмотрении задач приближенного аналитического решения краевых задач для дифференциальных уравнений [Sharan и др., 1997; Lagaris и др., 1998; Коваленко и др., 2017; Raissi и др., 2017; Raissi и др., 2019; Cai и др., 2021]. Изучались также подходы и способы учета начальных и граничных условий в виде особого функционала [Galperin и др., 1993; Galperin, Zheng, 1993].

Ключевой особенностью нейросетевого подхода является процесс самообучения на точно известных значениях и данных или при выполнении заранее определенных условий, записанных в виде систем дифференциальных и алгебраических уравнений. Идея использовать при обучении ИНС уравнения, описывающие физические явления, возникла сравнительно недавно. В последнее время в различных областях физики активно исследуются подходы для корректного решения подобного рода задач [Carleo и др., 2019; Karniadakis и др., 2021; Thuerey и др., 2022]. Модели, описывающие поведение физических систем, называют физико-информированными нейронными сетями (physics-informed neural networks, PINN). Практика применения PINN показывает, что на данный момент нет единого способа аппроксимации решений краевых задач. Выбор типа сети, ее структуры и методов обучения обычно определяется видом уравнения, свойствами аппроксимируемой функции и геометрией рассматриваемой задачи. Можно выделить два наиболее распространенных подхода: 1) аппроксимацию с помощью нейронных RBF-сетей, 2) аппроксимацию с помощью ИНС персептронного типа. В задачах с гладкими функциями успешно применяются RBF-сети. Применение персептронов, как правило,

оказывается оправданным, когда в задаче присутствуют резкие переходы или если есть основания ожидать, что такие переходы присутствуют в решении. Это может происходить, например, в случае разрывных функций, задающих коэффициенты уравнений, краевые, начальные или другие условия. В работах [Kansa, 1990a, 1990b; Kansa, 1999; Rasmussen, Williams, 2005; Васильев и др., 2015] рассмотрены алгоритмы аппроксимации решений краевых задач для уравнения Лапласа на плоскости с помощью RBF-сетей. В исследовании [Зрелова, Ульянов, 2022] описываются алгоритмы применения PINN перцептронного типа при решении задач классической механики. Динамика системы при этом может описываться уравнениями Лагранжа или Гамильтона. Авторы рассматривают применение гамильтоновых нейронных сетей как способ изучения гамильтоновой динамики. Решения уравнений Гамильтона позволяют предсказывать будущее поведение системы по ее текущему состоянию в фазовом пространстве. Обучение ИНС перцептронного типа осуществляется с использованием функции ошибок (функционала), составленной на основе уравнений Гамильтона. В статье [Cai и др., 2021] исследованы подходы к решению задач теплопереноса на основе полносвязной ИНС перцептронного типа. Рассмотрены результаты аппроксимации решений двух задач теплопереноса в различных типах геометрии, а именно: 1) задачи принудительной конвекции в закрытом помещении и 2) задачи обтекания цилиндра. Начальные и граничные условия учитываются в итоговом функционале ошибки обучения. Итоговый функционал обучения состоит из трех, учитывающих выполнимость, слагаемых: 1) уравнения теплопроводности, 2) условий на границе рассчитываемой области и 3) начальных условий. Результаты аппроксимации имеют высокую сходимость с численными решениями.

В данной работе подробно рассмотрена методика применения физико-информированных нейронных сетей перцептронного типа для уравнений произвольного типа и описаны способы обучения ИНС при решении начально-краевых задач для обыкновенных дифференциальных уравнений и уравнений в частных производных. В работе предлагаются авторские модификации функционала обучения нейронной сети для упрощения процедуры аппроксимации решений дифференциальных уравнений параболического типа.

Методы

Концепция применения нейронных сетей

Нейросетевое моделирование при решении краевых задач — это нахождение приближенного аналитического решения методами машинного обучения. В концепции применения нейронных сетей искомую неизвестную функцию аппроксимируют с помощью нейросетевой функции, которая определяется архитектурой используемой ИНС, видом функции активации и значениями весовых коэффициентов, соответствующих связям между узлами ИНС. Архитектура ИНС определяет порядок и последовательность действий; весовые коэффициенты связей определяют результат суперпозиции входных значений, поступающих на обработку в нейроны; функция активации определяет результат преобразования сигнала в каждом нейроне. Итоговое аппроксимирующее

значение вычисляется в результате многократного преобразования входного вектора при последовательном нахождении взвешенных значений (линейных суперпозиций) и вычислении функций активаций нейронов. Фактически ИНС реализует сложную функцию с несколькими уровнями вложения, содержащую в себе весовые коэффициенты-параметры. Настраиваемые весовые коэффициенты модели подбираются в процессе обучения исходя из условия минимизации функционала, характеризующего качество аппроксимации. Таким образом, процесс обучения нейронной сети — это процесс нахождения наилучшей аппроксимации искомого решения сложной нейросетевой функцией. Как показано в работе [Васильев и др., 2015], сложная многоуровневая функция, построенная даже из простейших полиномов первого и второго порядка, способна аппроксимировать решение дифференциальных уравнений на широком интервале аргумента. Одной из простых нейросетевых моделей является модель персептрона. Нейросетевое решение для двухслойного персептрона (с одним скрытым слоем) может быть описано функцией:

$$y = W_y \cdot \text{th}(W_x \cdot x + b_x) + b_y, \quad (1)$$

где x — входной вектор данных, который подается на вход нейросети; th — тангенциальная функция активации нейронов; $W_x(n, m)$, $W_y(m, 1)$, $b_x(1, m)$, $b_y(1, 1)$ — матрицы настроечных коэффициентов (массивы весов модели); n — количество входных параметров модели; m — количество нейронов модели; операция « \cdot » — матричное умножение.

Большинство физических законов формулируется в виде дифференциальных уравнений. Для расчета поведения физических объектов в этом случае требуется решать начально-краевые задачи. Решением начально-краевой задачи для дифференциального уравнения вида

$$F\left(x_1, x_2, \dots, x_m, z, \frac{\partial z}{\partial x_1}, \frac{\partial z}{\partial x_2}, \dots, \frac{\partial z}{\partial x_m}, \frac{\partial^2 z}{\partial x_1^2}, \frac{\partial^2 z}{\partial x_1 \partial x_2}, \frac{\partial^2 z}{\partial x_m^2}\right) = 0 \quad (2)$$

является функция $z(x_1, x_2, \dots, x_m)$, для которой при подстановке в исходное уравнение выполняется тождественное равенство правой и левой части. Кроме того, эта функция-решение должна удовлетворять граничным и начальным условиям. Тогда функцию (функционал) ошибки обучения для дифференциального уравнения (2) можно записать в виде:

$$\begin{aligned} \text{loss} = & F\left(x_1, x_2, \dots, x_m, z, \frac{\partial z}{\partial x_1}, \frac{\partial z}{\partial x_2}, \dots, \frac{\partial z}{\partial x_m}, \frac{\partial^2 z}{\partial x_1^2}, \frac{\partial^2 z}{\partial x_1 \partial x_2}, \frac{\partial^2 z}{\partial x_m^2}\right)^2 + \\ & + \sum_{\text{НУ и ГУ}} (z_{\text{fact}} - z_{\text{calc}})^2, \end{aligned} \quad (3)$$

где z_{fact} , z_{calc} — заданные и вычисленные значения искомой функции в точках на граничных и начальных условиях.

Производные функции $\partial z/\partial x_1$ могут быть найдены в явном виде через функции активации нейронов или с использованием библиотеки автоматического дифференцирования. Нахождение весовых коэффициентов осуществляется с помощью стандартных оптимизационных алгоритмов обучения нейронной сети.

Результаты и обсуждение

Решение задачи Коши для обыкновенного дифференциального уравнения первого порядка

Применим описанный выше подход для решения задачи Коши обыкновенного дифференциального уравнения, которая имеет аналитическое решение (4):

$$\frac{dT}{dt} = -T(t) \cdot t, \quad T(-2) = 1, \quad T(t) = e^{\frac{4-t^2}{2}}. \quad (4)$$

Аппроксимацию решения задачи (4) будем искать на промежутке $t \in [-2; 2]$ с использованием модели двухслойного персептрона. Количество нейронов на промежуточном слое (гиперпараметр, подбираемый эмпирически) равно 20. Для обучения ИНС будем использовать равноотстоящие точки, делящие отрезок $[-2; 2]$ на равные части. Рассмотрим два случая: 1) обучение на 100 точках и 2) обучение на разреженном множестве из 50 точек. Функция ошибки обучения запишется в виде:

$$L = \sum_{i=1}^n \left(\frac{dT}{dt} + T(t) \right)^2 + (T_{t=0 \text{ fact}} - T_{t=0 \text{ calc}})^2 \rightarrow 0. \quad (5)$$

Аппроксимация исходной функции $T(t)$ осуществляется с помощью модели многослойного персептрона с одним промежуточным слоем:

$$T(t) = W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y. \quad (6)$$

Производная функции $T(t)$ запишется в виде:

$$\frac{dT}{dt} = W_y \cdot \text{th}'(W_x \cdot (t) + b_x) \cdot W_x. \quad (7)$$

Тогда, подставив выражение (7) в функционал обучения (5), получим:

$$L = \sum_{i=1}^n \left(W_y \cdot \text{th}'(W_x \cdot (t) + b_x) \cdot W_x + (W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y) \cdot t \right)^2 + (T_{t=0 \text{ fact}} - (W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y)_{t=0 \text{ calc}})^2 \rightarrow 0. \quad (8)$$

Поиск оптимальных весовых коэффициентов для минимизации функционала (8) производился с помощью итерационного численного оптимизационного алгоритма — модифицированного градиентного спуска Adam. Правило обновления весов алгоритма Adam представлено выражением (9) [Kingma, Ba, 2015; Хайкин, 2019]:

$$W_x(t+1) = W_x(t) - l_{step} \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}}, \quad (9)$$

где

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \widehat{v}_t = \frac{v_t}{1 - \beta_2^t}, \quad m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \frac{\partial L}{\partial W_x},$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \frac{\partial L^2}{\partial W_x}.$$

Adam является популярным алгоритмом в области глубокого обучения. Практика показывает, что Adam выгодно отличается от других методов стохастической оптимизации за счет реализации в себе двух возможностей: 1) накопления движения градиентов в ходе обучения и 2) более слабого обновления весов для типичных признаков.

Для расчета алгоритма оптимизации Adam требуется найти производные функции $\partial L / \partial W_x, \partial L / \partial W_y, \partial L / \partial b_x, \partial L / \partial b_y$. Данные производные можем определить в явном виде, дифференцируя функционал (8):

$$\begin{aligned} \frac{\partial L}{\partial W_y} = & \sum 2 \cdot (\text{th}'(W_x \cdot (t) + b_x) \cdot W_x + (\text{th}(W_x \cdot (t) + b_x) + b_y) \cdot t) - \\ & - 2 \cdot (T_{t=0 \text{ fact}} - (W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y)_{t=0 \text{ calc}}) \times \\ & \times \text{th}(W_x \cdot (x, t) + b_x), \end{aligned} \quad (10)$$

$$\begin{aligned} \frac{\partial L}{\partial W_x} = & \sum 2 \cdot (W_y \cdot \text{th}'(W_x \cdot (t) + b_x) + W_y \cdot \text{th}''(W_x \cdot (t) + b_x) \cdot W_x \cdot x) - \\ & - 2 \cdot (T_{t=0 \text{ fact}} - (W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y)_{t=0 \text{ calc}}) \times \\ & \times W_y \cdot \text{th}'(W_x \cdot (t) + b_x) \cdot x, \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{\partial L}{\partial b_x} = & \sum 2 \cdot (W_y \cdot \text{th}''(W_x \cdot (t) + b_x) \cdot W_x) - \\ & - 2 \cdot (T_{t=0 \text{ fact}} - (W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y)_{t=0 \text{ calc}}) \times \\ & \times W_y \cdot \text{th}'(W_x \cdot (t) + b_x), \end{aligned} \quad (12)$$

$$\frac{\partial L}{\partial b_y} = \sum -2 \cdot (T_{t=0 \text{ fact}} - (W_y \cdot \text{th}(W_x \cdot (t) + b_x) + b_y)_{t=0 \text{ calc}}). \quad (13)$$

Обучение проводилось с длительностью 1 000 эпох (итераций). Важным параметром алгоритма оптимизации Adam является шаг обновления весовых коэффициентов. График поведения функции ошибок L при двух разных значениях параметра l_{step} (шаг оптимизационного алгоритма) представлен на рис. 1.

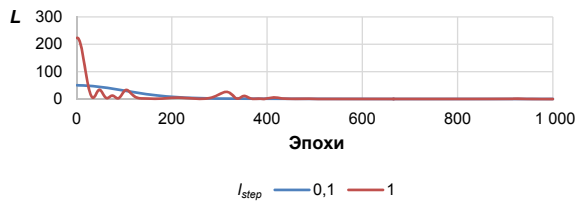


Рис. 1. Поведение функции ошибок L при разных шагах обучения l_{step} при аппроксимации решения задачи Коши (4) для обыкновенного дифференциального уравнения

Fig. 1. The behavior of the error function L at different learning steps l_{step} when approximating the solution of the Cauchy problem (4) for an ordinary differential equation

Можно заметить, что при большом шаге оптимизатора ($l_{step} = 1$) поиск решения становится менее стабильным на ранних этапах обучения, но тем не менее сходится к значению, близкому к нулю.

Сопоставление точного аналитического решения и решений, полученных с помощью нейросетевой аппроксимации, представлено на рис. 2.

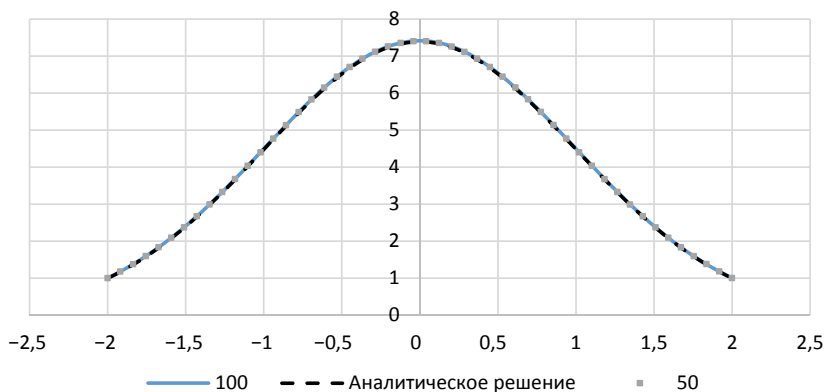


Рис. 2. Графики аналитического и нейросетевых решений (полученных при обучении на 100 и 50 точках) задачи Коши (4) для обыкновенного дифференциального уравнения

Fig. 2. Graphs of analytical and neural network solutions (obtained by training at 100 and 50 points) of the Cauchy problem (4) for an ordinary differential equation

Численные значения аппроксимации совпадают с точностью до 0,3%, что говорит о высоком качестве аппроксимации аналитической функции методом ИНС. Также стоит отметить совпадение качества полученной аппроксимации как на редкой сетке точек (50 точек), так и на уплотненной сетке (100 точек). То есть можно предположить, что при использовании нейросетевого моделирования удастся получить достаточно точную аппроксимацию решения без чрезмерного измельчения области допустимых значений (ОДЗ).

Решение начально-краевой задачи для дифференциальных уравнений в частных производных параболического типа

Рассмотрим более сложную задачу — решение начально-краевой задачи для одномерного уравнения теплопроводности:

$$\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2}, \quad (14)$$

где $a = 1 \text{ м}^2/\text{с}$, $x \in [0; 100]$, $t \in [0; 100]$.

Начальные и граничные условия с разрывом имеют вид:

$$T_{t=0} = 1, \quad T_{x=0} = 0, \quad T_{x=100} = 1. \quad (15)$$

Алгоритм поиска аппроксимации принципиально не отличается от приведенного выше алгоритма решения задачи Коши обыкновенного дифференциального уравнения. Отличия проявляются в более сложной форме функционала обучения и, как следствие, более сложном и ресурсоемком процессе обучения нейронной сети.

Рассмотрим аппроксимацию исходной функции $T(x, t)$ с помощью модели многослойного перцептрона с одним промежуточным слоем:

$$T(x, t) = W_y \cdot \text{th}(W_x \cdot (x, t) + b_x) + b_y. \quad (16)$$

Тогда частные производные определяются следующими выражениями:

$$\frac{\partial T}{\partial t} = W_y \cdot \text{th}'(W_x \cdot (x, t) + b_x) \cdot W_x(2), \quad (17)$$

где (2) — номер строчки матрицы весовых коэффициентов W_x при координате t ;

$$\frac{\partial^2 T}{\partial x^2} = W_y \cdot \text{th}''(W_x \cdot (x, t) + b_x) \cdot W_x^2(1), \quad (18)$$

где (1) — номер строчки матрицы весовых коэффициентов W_x при координате x .

Подставим выражения (17) и (18) в исходное уравнение (14). Запишем целевой функционал ошибки обучения для аппроксимации уравнения (14):

$$W_y \cdot \text{th}'(W_x \cdot (x, t) + b_x) \cdot W_x(2) - a \cdot W_y \cdot \text{th}''(W_x \cdot (x, t) + b_x) \cdot W_x^2(1) \rightarrow 0. \quad (19)$$

Упростим данное выражение: несложно заметить, что выражение можно сократить на матрицу весовых коэффициентов W_y . Производные гиперболического тангенса также можно представить в виде выражений:

$$\text{th}'(x) = (1 - \text{th}^2(x)), \quad (20)$$

$$\text{th}''(x) = 2 \cdot (\text{th}^2(x) - 1) \cdot \text{th}(x). \quad (21)$$

С учетом выражений (20) и (21) выражение (19) после сокращения на W_y примет вид:

$$\begin{aligned} & (1 - \text{th}^2 (W_x \cdot (x, t) + b_x)) \cdot W_x(2) - \\ & - a \cdot 2 \cdot (\text{th}^2 (W_x \cdot (x, t) + b_x) - 1) \times \\ & \times \text{th} (W_x \cdot (x, t) + b_x) \cdot W_x^2(1) \rightarrow 0. \end{aligned} \quad (22)$$

Сократив (22) на выражение $(1 - \text{th}^2 (W_x \cdot (x, t) + b_x))$, получим упрощенный функционал обучения для аппроксимации решения дифференциального уравнения:

$$W_x(2) + a \cdot 2 \cdot \text{th} (W_x \cdot (x, t) + b_x) \cdot W_x^2(1) \rightarrow 0. \quad (23)$$

Анализируя выражение (23), можно сделать вывод, что для улучшения точности аппроксимации решения задачи (14) нужно изменять только весовые коэффициенты W_x первого слоя нейронной сети, т. е. коэффициенты при входном векторе. Коэффициенты второго слоя не влияют на функционал обучения (23), но они используются для настройки итоговой аппроксимации на граничные и начальные условия.

Итоговый функционал обучения (с учетом краевой задачи) записывается следующим образом:

$$\begin{aligned} & \sum_{i=1}^n W_x(2) + a \cdot 2 \cdot \text{th} (W_x \cdot (x, t) + b_x) \cdot W_x^2(1) + \\ & + \sum_x (T(x, 0) - T(x, t))^2 + (T(0, t) - T(x, t))^2 \rightarrow 0. \end{aligned} \quad (24)$$

Поиск оптимальных весовых коэффициентов производился с помощью оптимизационного алгоритма Adam, описанного выше. Производные L -функции $\partial L / \partial W_x$, $\partial L / \partial W_y$, $\partial L / \partial b_x$, $\partial L / \partial b_y$ определены через упрощенный функционал (24).

Как было показано ранее, коэффициенты второго слоя нейронной сети не влияют на сходимость к решению дифференциального уравнения. Следовательно, производные данных весов будут рассчитываться только для второго слагаемого, описывающего сходимость к начальным и граничным условиям:

$$\frac{\partial L}{\partial W_y} = \sum -2 \cdot (T_{fact} - T(x, t)) \cdot \text{th} (W_x \cdot (x, t) + b_x), \quad (25)$$

$$\frac{\partial loss}{\partial b_y} = \sum -2 \cdot (T_{fact} - T(x, t)). \quad (26)$$

Производные первого слоя нейронной сети зависят от двух слагаемых функционала (24). Следовательно, итоговую производную можно разбить на две составляющие.

1. Производные от функционала дифференциального уравнения:

$$\frac{\partial L}{\partial W_x(2)} = \sum 2 \cdot loss' \cdot (1 + 2 \cdot a \cdot W_x^2(1) \cdot \text{th}' (W_x \cdot (x, t) + b_x) \cdot t), \quad (27)$$

$$\frac{\partial L}{\partial W_x(1)} = \sum 2 \cdot \text{loss}' \cdot \left(2 \cdot a \cdot W_x^2(1) \cdot \text{th}''(W_x \cdot (x, t) + b_x) \cdot x + \right), \quad (28)$$

$$\frac{\partial L}{\partial b_x} = \sum 2 \cdot \text{loss}' \cdot (2 \cdot a \cdot W_x^2(1) \cdot \text{th}''(W_x \cdot (x, t) + b_x)). \quad (29)$$

2. Производные от функционала начальных и граничных условий:

$$\frac{\partial L}{\partial W_x} = \sum -2 \cdot (T_{fact} - T(x, t)) \cdot W_y \cdot \text{th}'(W_x \cdot (x, t) + b_x) \cdot (x, t), \quad (30)$$

$$\frac{\partial L}{\partial b_x} = \sum -2 \cdot (T_{fact} - T(x, t)) \cdot W_y \cdot \text{th}'(W_x \cdot (x, t) + b_x). \quad (31)$$

Для расчета итогового градиента обновления весов W_x нужно построчно просуммировать элементы, определенные выражениями (30), (28) и (27). Аналогично для весов b_x нужно просуммировать выражения (31) и (29).

Обучение нейронной сети проводилось на разреженной сетке 20×20 в координатах (x, t) . График поведения L -функции при различных шагах обучения приведен на рис. 3.

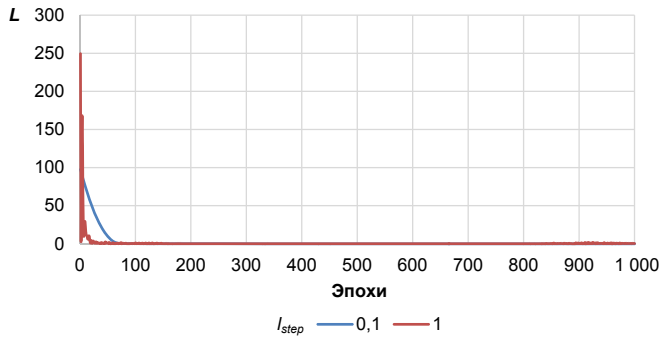


Рис. 3. Поведение функции ошибок L при разных шагах обучения l_{step} при аппроксимации решения начально-краевой параболического дифференциального уравнения

Fig. 3. Behavior of the error function L at different learning steps l_{step} when approximating the solution of the initial-boundary parabolic differential equation

При анализе графиков поведения L -функции (рис. 3) можно заметить, что стабильный минимум достигается примерно на 40 эпох быстрее при $l_{step} = 1$, чем при $l_{step} = 0,1$.

Для оценки качества аппроксимации нейросетевой модели было найдено численное решение задачи (5)–(6) конечно-разностным методом по неявной схеме с применением метода Якоби для решения получившейся системы линейных уравнений. Сетка поиска сеточного решения имела 100 равноотстоящих слоев по координате x и 100 слоев по времени t . Обучение нейронной сети производилось на разреженной сетке 20×20 . Сопоставление полученных решений в различные моменты времени в узлах обучающей сетки 20×20 представлено на рис. 4.

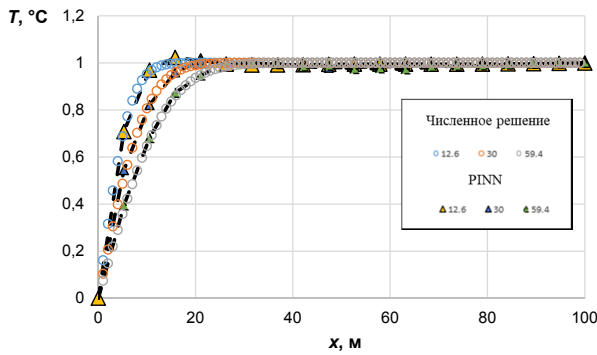


Рис. 4. Сравнение численного и нейросетевого решений уравнения теплопроводности в различные моменты времени t (12,6, 30 и 60 с)

Fig. 4. Comparison of numerical and neural network solutions of the thermal conductivity equation at different time points t (12.6, 30 and 60 s)

Также были построены поверхности решений (рис. 5). Для этого в обученную нейронную сеть были поданы координаты точек по уплотненной сетке 100×100 и рассчитаны значения температуры T в узлах новой сетки без переобучения модели.

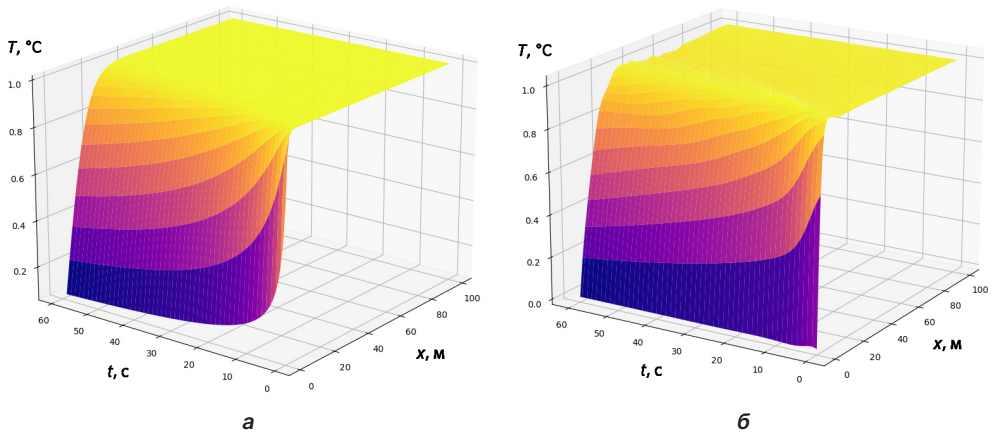


Рис. 5. Сравнение численного (а) и нейросетевого (б) решений уравнения теплопроводности в различные моменты времени t

Fig. 5. Comparison of numerical (a) and neural network (б) solutions of the thermal conductivity equation at different time points t

Сравнение решений, найденных конечно-разностным методом и с использованием обученной ИНС в точках уплотненной сетки, не участвовавших в обучении, в различные моменты времени, представлено на рис. 6.

В среднем рассчитанные значения на уплотненной сетке отличаются не более чем на 1,7 %, что также свидетельствует о высоком качестве аппроксимации исходного решения методом ИНС с применением разреженной сетки точек обучения.

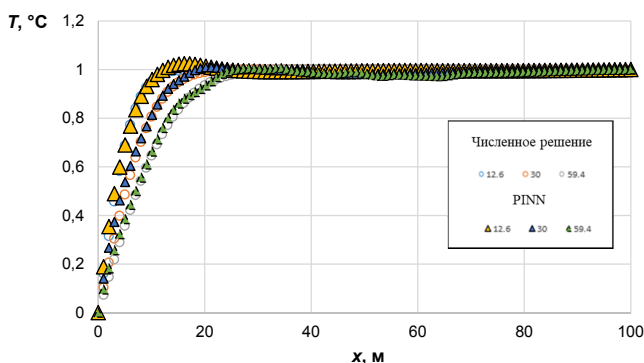


Рис. 6. Сравнение решений, найденных конечно-разностным методом и с использованием обученной ИНС в точках уплотненной сетки в различные моменты времени t (12,6, 30 и 60 с)

Fig. 6. Comparison of solutions found by the finite-difference method and using the trained INS at points of the compacted grid at different time points t (12.6, 30 and 60 s)

Заключение

1. Нейросетевое моделирование на основе ИНС перцептронного типа с одним скрытым слоем позволяет получать аналитическую аппроксимацию решения начально-краевых задач для обыкновенных дифференциальных уравнений и уравнений в частных производных параболического типа, в том числе с разрывом на границе.
2. Обучение на разреженной сетке точек показало высокую точность аппроксимации решения рассмотренных краевых задач, что позволяет минимизировать затраты машинного времени на процесс обучения.
3. Рассмотренный подход может быть использован для решения задач с произвольными начальными и граничными условиями. Усложнение формы ОДЗ и вида граничных условий может потребовать использования более сложной архитектуры ИНС.
4. Использование нейросетевого подхода при решении краевых задач для дифференциальных уравнений позволяет расширить класс численно решаемых задач, включив в него задачи поиска решений уравнений с условиями, определенными как на внешней границе, так и в отдельных точках ОДЗ. Этот подход позволит решать задачи адаптации параметрических математических моделей на фактические показатели.

Список источников

- Васильев А. Н., Тархов Д. А., Шемякина Т. А. 2015. Нейросетевой подход к задачам математической физики. СПб.: Нестор-История. 260 с.
- Зрелова Д. П., Ульянов С. В. 2022. Модели физически информированных / осведомленных классических Лагранжевых / Гамильтоновых нейронных сетей в глубоком обучении //

- Современные информационные технологии и ИТ-образование. Том 18. № 2. С. 310–325. <https://doi.org/10.25559/SITITO.18.202202.310-325>
- Коваленко А. Н., Черноморец А. А., Петина М. А. 2017. О применении нейронных сетей для решения дифференциальных уравнений в частных производных // Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. № 9 (258). С. 103–110.
- Колмогоров А. Н. 1956. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Доклады АН СССР. Том 108. № 2. С. 179–182.
- Тархов Д. А. 2014. Нейросетевые модели и алгоритмы. М.: Радиотехника. 348 с.
- Хайкин С. 2019. Нейронные сети. М.; СПб.: Диалектика. 1103 с.
- Cai S., Wang Z., Wang S., Perdikaris P., Karniadakis G. E. 2021. Physics-informed neural networks for heat transfer problems // Journal of Heat Transfer. Vol. 143. No. 6. Article 060801. <https://doi.org/10.1115/1.4050542>
- Carleo G., Cirac I., Cranmer K., Daudet L., Schuld M., Tishby N., Vogt-Maranto L., Zdeborová L. 2019. Machine learning and the physical sciences // Reviews of Modern Physics. Vol. 91. No. 4. Article 045002. <https://doi.org/10.1103/RevModPhys.91.045002>
- Galperin E. A., Pan Z., Zheng Q. 1993. Application of global optimization to implicit solution of partial differential equations // Computers & Mathematics with Applications. Vol. 25. No. 10–11. Pp. 119–124. [https://doi.org/10.1016/0898-1221\(93\)90287-6](https://doi.org/10.1016/0898-1221(93)90287-6)
- Galperin E. A., Zheng Q. 1993. Solution and control of PDE via global optimization methods // Computers & Mathematics with Applications. Vol. 25. No. 10–11. Pp. 103–118. [https://doi.org/10.1016/0898-1221\(93\)90286-5](https://doi.org/10.1016/0898-1221(93)90286-5)
- Kansa E. J. 1990a. Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — I surface approximations and partial derivative estimates // Computers & Mathematics with Applications. Vol. 19. No. 8. Pp. 127–145. [https://doi.org/10.1016/0898-1221\(90\)90270-T](https://doi.org/10.1016/0898-1221(90)90270-T)
- Kansa E. J. 1990b. Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — II solutions to parabolic, hyperbolic and elliptic partial differential equations // Computers & Mathematics with Applications. Vol. 19. No. 8. Pp. 147–161. [https://doi.org/10.1016/0898-1221\(90\)90271-K](https://doi.org/10.1016/0898-1221(90)90271-K)
- Kansa E. J. 1999. Motivation for using radial basis functions to solve PDEs. Lawrence Livermore National Laboratory; Embry-Riddle Aeronautical University.
- Karniadakis G. E., Kevrekidis I. G., Lu L., Perdikaris P., Wang S., Yang L. 2021. Physics-informed machine learning // Nature Reviews Physics. Vol. 3. No. 6. Pp. 422–440. <https://doi.org/10.1038/s42254-021-00314-5>
- Kingma D. P., Ba J. 2015. Adam: A method for stochastic optimization // The 3rd International Conference for Learning Representations (7–9 May 2015, San Diego, CA, USA). <https://doi.org/10.48550/arXiv.1412.6980>
- Lagaris I. E., Likas A., Fotiadis D. I. 1998. Artificial neural networks for solving ordinary and partial differential equations // IEEE Transactions on Neural Networks. Vol. 9. No. 5. Pp. 987–1000. <https://doi.org/10.1109/72.712178>
- Raissi M., Perdikaris P., Karniadakis G. E. 2017. Machine learning of linear differential equations using Gaussian processes // Journal of Computational Physics. Vol. 348. Pp. 683–693. <https://doi.org/10.1016/j.jcp.2017.07.050>

- Raissi M., Perdikaris P., Karniadakis G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations // *Journal of Computational Physics*. Vol. 378. Pp. 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Rasmussen C. E., Williams C. K. I. 2005. *Gaussian Processes for Machine Learning*. The MIT Press. <https://doi.org/10.7551/mitpress/3206.001.0001>
- Sharan M., Kansa E. J., Gupta S. 1997. Application of the multiquadric method for numerical solution of elliptic partial differential equations // *Applied Mathematics and Computation*. Vol. 84. No. 2. Pp. 275–302. [https://doi.org/10.1016/S0096-3003\(96\)00109-9](https://doi.org/10.1016/S0096-3003(96)00109-9)
- Thuerey N., Holl Ph., Mueller M., Schnell P., Trost F., Um K. 2022. Physics-based Deep Learning. <https://doi.org/10.48550/arXiv.2109.05237>

References

- Vasiliev, A. N., Tarhov, D. A., & Shemyakina, T. A. (2015). *Neural network approach to problems of mathematical physics*. Nestor-History. [In Russian]
- Zrelova, D. P., & Ulyanov, S. V. (2022). Physics-informed classical Lagrange / Hamilton neural networks in deep learning. *Modern Information Technologies and IT-Education*, 18(2), 310–325. <https://doi.org/10.25559/SITITO.18.202202.310-325> [In Russian]
- Kovalenko, A. N., Chernomorets, A. A., & Petina, M. A. (2017). On the neural networks application for solving of partial differential equations. *Scientific Bulletin of Belgorod State University. Series: Economics. Informatics*, (9), 103–110. [In Russian]
- Kolmogorov, A. N. (1956). On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. *Proceedings of the USSR Academy of Sciences*, 108(2), 179–182. [In Russian]
- Tarhov, D. A. (2014). *Neural network models and algorithms*. Radiotekhnika. [In Russian]
- Haykin, S. (2019). *Neural networks*. Dialektika. [In Russian]
- Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6), Article 060801. <https://doi.org/10.1115/1.4050542>
- Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), Article 045002. <https://doi.org/10.1103/RevModPhys.91.045002>
- Galperin, E. A., Pan, Z., & Zheng, Q. (1993). Application of global optimization to implicit solution of partial differential equations. *Computers & Mathematics with Applications*, 25(10–11), 119–124. [https://doi.org/10.1016/0898-1221\(93\)90287-6](https://doi.org/10.1016/0898-1221(93)90287-6)
- Galperin, E. A., & Zheng, Q. (1993). Solution and control of PDE via global optimization methods. *Computers & Mathematics with Applications*, 25(10–11), 103–118. [https://doi.org/10.1016/0898-1221\(93\)90286-5](https://doi.org/10.1016/0898-1221(93)90286-5)
- Kansa, E. J. (1990a). Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — I surface approximations and partial derivative estimates. *Computers & Mathematics with Applications*, 19(8), 127–145. [https://doi.org/10.1016/0898-1221\(90\)90270-T](https://doi.org/10.1016/0898-1221(90)90270-T)
- Kansa, E. J. (1990b). Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics — II solutions to parabolic, hyperbolic and elliptic partial

- differential equations. *Computers & Mathematics with Applications*, 19(8), 147–161. [https://doi.org/10.1016/0898-1221\(90\)90271-K](https://doi.org/10.1016/0898-1221(90)90271-K)
- Kansa, E. J. (1999). *Motivation for using radial basis functions to solve PDEs*. Lawrence Livermore National Laboratory; Embry-Riddle Aeronautical University.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440. <https://doi.org/10.1038/s42254-021-00314-5>
- Kingma, D. P., & Ba, J. (2015, May 7–9). *Adam: A method for stochastic optimization* [Conference paper]. The 3rd International Conference for Learning Representations, San Diego, CA, USA. <https://doi.org/10.48550/arXiv.1412.6980>
- Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5), 987–1000. <https://doi.org/10.1109/72.712178>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348, 683–693. <https://doi.org/10.1016/j.jcp.2017.07.050>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. The MIT Press. <https://doi.org/10.7551/mitpress/3206.001.0001>
- Sharan, M., Kansa, E. J., & Gupta, S. (1997). Application of the multiquadric method for numerical solution of elliptic partial differential equations. *Applied Mathematics and Computation*, 84(2), 275–302. [https://doi.org/10.1016/S0096-3003\(96\)00109-9](https://doi.org/10.1016/S0096-3003(96)00109-9)
- Thuerey, N., Holl, Ph., Mueller, M., Schnell, P., Trost, F., & Um, K. (2022). *Physics-based deep learning*. <https://doi.org/10.48550/arXiv.2109.05237>

Информация об авторах

Владимир Евгеньевич Вершинин, главный специалист, Тюменский нефтяной научный центр, Тюмень, Россия
ve_vershinin2@tnnc.rosneft.ru

Роман Юрьевич Пономарев, главный специалист, Тюменский нефтяной научный центр, Тюмень, Россия
ryponomarev@tnnc.rosneft.ru

Information about the authors

Vladimir E. Vershinin, Chief Specialist, Tyumen Petroleum Research Center, Tyumen, Russia
ve_vershinin2@tnnc.rosneft.ru

Roman Yu. Ponomarev, Chief Specialist, Tyumen Petroleum Research Center, Tyumen, Russia
ryponomarev@tnnc.rosneft.ru