

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Г.А. Маргарян¹

¹Самарский университет, г. Самара
e-mail: 2020-00948@students.ssau.ru

Аннотация

В работе представлена нейронная сеть прямого распространения для решения обыкновенных дифференциальных уравнений высших порядков. Основное внимание уделено модификации сети для предотвращения её переобучения. При сравнении полученного решения с методами Эйлера и Рунге-Кутты наблюдалось уменьшение погрешности от 20 до 60 раз. Дополнительно выполнен анализ вычислительных затрат, устойчивости к шуму и масштабируемости.

Ключевые слова: обыкновенные дифференциальные уравнения, нейронная сеть, метод коллокаций, метод Эйлера, погрешность, метод Рунге–Кутты, регуляризация.

Введение

Численное решение дифференциальных уравнений является фундаментальной задачей вычислительной математики. Традиционные подходы обладают рядом ограничений: они предоставляют дискретное решение в узлах сетки и требуют значительных вычислительных ресурсов при её измельчении. В связи с этим активно используются искусственные нейронные сети [1]. Так, в работе [2] предложен метод решения дифференциальных уравнений с использованием полносвязной нейронной сети. Рассмотрен многослойный перцептрон с одним скрытым слоем, содержащим десять нейронов с сигмоидальной функцией активации. Эффективность подхода была продемонстрирована на задачах, включающих ОДУ первого и второго порядков и нелинейные уравнения в частных производных. Однако, как отмечают авторы [2], использованная архитектура оставалась фиксированной, а влияние количества скрытых слоёв на качество решения задач высших порядков не исследовалось. Таким образом целью настоящей работы является восполнение указанного пробела.

1. Нейронная сеть без оптимизации

В [2] представлено дифференциальное уравнение вида:

$$G(\vec{x}, \psi(\vec{x}), \nabla\psi(\vec{x}), \nabla^2\psi(\vec{x})) = 0, \quad \vec{x} \in D, \quad (1)$$

с граничными условиями (например, Дирихле или Неймана), где: $\vec{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ – вектор независимых переменных, $\psi(\vec{x})$ – искомая функция, а $D \subset \mathbb{R}^n$ – область определения искомой функции $\psi(\vec{x})$. Для численного решения (1) в [2] предлагается использовать метод коллокаций [3], который заключается в дискретизации области D и ее границы S на множество точек $\hat{D}$ и $\hat{S}$ соответственно. Это, согласно [2] преобразует задачу в систему уравнений:

$$G(\vec{x}_i, \psi(\vec{x}_i), \nabla \psi(\vec{x}_i), \nabla^2 \psi(\vec{x}_i)) = 0, \forall \vec{x}_i \in \hat{D}. \quad (2)$$

Если обозначить через $\psi_i(\vec{x}, \vec{p})$ пробное решение с настраиваемыми параметрами $\vec{p}$, то задача сводится к безусловной оптимизации уравнения:

$$\min_{\vec{p}} \sum_{\vec{x}_i \in \hat{D}} [G(\vec{x}_i, \psi_t(\vec{x}_i, \vec{p}), \nabla \psi_t(\vec{x}_i, \vec{p}), \nabla^2 \psi_t(\vec{x}_i, \vec{p}))]^2. \quad (3)$$

Пробное решение представляется в виде суммы двух слагаемых:

$$\psi_t(\vec{x}) = A(\vec{x}) + F(\vec{x}, N(\vec{x}, \vec{p})). \quad (4)$$

Первое слагаемое, $A(\vec{x})$, не содержит настраиваемых параметров и полностью определяется граничными условиями, гарантируя их выполнение. Второе слагаемое, F включает нейронную сеть прямого распространения $N(\vec{x}, \vec{p})$, которая обучается так, чтобы минимизировать невязку дифференциального уравнения во внутренних точках области. В работе используется многослойный перцептрон с одним скрытым слоем. Количество нейронов во входном слое равно размерности задачи n . Скрытый слой содержит десять нейронов с сигмоидальной функцией активации $\sigma(z) = \frac{1}{1+e^{-z}}$, и это количество остаётся неизменным во всех экспериментах. Выходной слой состоит из одного линейного нейрона. Минимизацию невязки для уравнения (3) можно рассматривать как процедуру обучения нейронной сети, где ошибка, соответствующая каждому входному вектору $\vec{x}_i$ представляет собой значение $G(\vec{x}_i)$, которое должно стать нулевым.

2. Оптимизация нейронной сети

В настоящей работе зададим ОДУ третьего порядка [4]:

$$y''' + 2f(x)y' + f'(x)y = 0, \quad (5)$$

В ходе проведения вычислительных экспериментов выяснилось, что для уменьшения погрешности до приемлемой (в 20 раз по сравнению с методом из [2]), необходимо увеличить количество нейронов на скрытом слое до 30. Для обучения нейронной сети был использован алгоритм градиентного спуска с шагом равным $1e^{-3}$. Критерием остановки служило достижение заданного числа итераций (5000), ранняя остановка применялась только в случае получения неизмеримой величины ошибки (NaN). В качестве функции потерь использовалась сумма квадратов невязок в точках коллокации с добавлением L2-регуляризации, что позволило снизить риск переобучения. Инициализация весов нейронной сети выполнялась по методу Xavier/Glorot. Для более убедительной оценки предложенного подхода проведено сравнение не только с исходной архитектурой из [2] и методом Эйлера, но и с классическим методом Рунге-Кутты 4-го порядка, а также с альтернативной архитектурой нейронной сети, содержащей два скрытых слоя. Для обучения такой сети использовался оптимизатор Adam с dropout для предотвращения переобучения, в качестве критерия остановки обучения выступало достижение заданного числа итераций (5000). Подтвердилось предположение авторов работы [2] о недопустимости использования нейронной сети из [2] при решении обыкновенных дифференциальных уравнений высших порядков (рис. 1).

Кроме того, рассмотрению подлежали ОДУ четвертого порядка [4]:

$$y^{(4)} - 12y'' + 12y = 16x^4 \exp x^2. \quad (6)$$

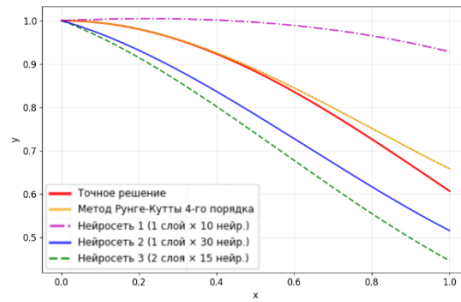


Рисунок 1 – Решение ОДУ третьего порядка

Главной проблемой при работе с уравнением (6) оказалось переобучение нейронной сети, для предотвращения которого был реализован механизм адаптации шага при вычислении производных через конечные разности. Параметры обучения для оптимизированной сети оставались прежними. Для проведения сравнения с классическим методом Рунге-Кутты 4-го порядка, а также с альтернативной архитектурой нейронной сети, содержащей два скрытых слоя, применялся метод градиентного спуска с автоматическим вычислением градиентов посредством библиотеки Autograd. Обучение проводилось со следующими гиперпараметрами: скорость обучения – $1e^{-3}$, количество точек коллокации – 20. Для двухслойной архитектуры использовались две функции активации: ReLU на первом скрытом слое и сигмоидальная – на втором. Остановка обучения производилась при достижении заданного числа итераций (5000). Таким образом при решении уравнения (6) применение авторской модификации нейронной сети из [2] позволило уменьшить погрешность в 60 раз (рис. 2). Сравнение точности и вычислительных затрат для решения уравнений (5) и (6) представлены в таблице 1.

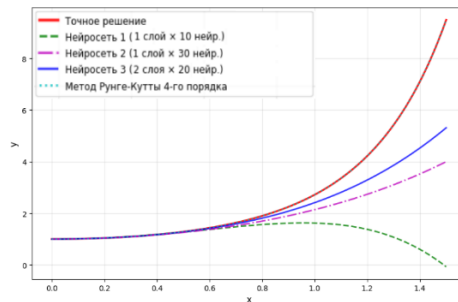


Рисунок 2 – Решение ОДУ четвертого порядка

Таблица 1 – Сравнение точности и вычислительных затрат

ОДУ 3-го порядка			ОДУ 4-го порядка		
Метод	Глоб/лок ошибка	Время, мин	Метод	Глоб/лок ошибка	Время, мин
Эйлер	$2.3 \cdot 10^{-2} / 5.3 \cdot 10^{-3}$	10	Эйлер	$2.34 \cdot 10^{-2} / 1.8 \cdot 10^{-3}$	10
Рунге-Кутте	$1.1 \cdot 10^{-4} / 2.7 \cdot 10^{-8}$	25	Рунге-Кутте	$4.12 \cdot 10^{-4} / 4.3 \cdot 10^{-8}$	25
НС 1	$4.2 \cdot 10^{-3} / 5.8 \cdot 10^{-4}$	45	НС 1	$3.45 \cdot 10^{-2} / 4.6 \cdot 10^{-4}$	45
НС 2	$2.1 \cdot 10^{-4} / 4.6 \cdot 10^{-5}$	60	НС 2	$5.75 \cdot 10^{-4} / 7.2 \cdot 10^{-5}$	142
НС 3	$1.9 \cdot 10^{-5} / 2.4 \cdot 10^{-8}$	135	НС 3	$2.25 \cdot 10^{-5} / 1.6 \cdot 10^{-8}$	168

Нейросетевой подход требует большего времени из-за итерационного обучения. Однако он предоставляет решение в аналитической форме и легко адаптируется к нерегулярным сеткам. Для реальных задач, где допустимы высокие вычислительные затраты нейросетевой метод оправдан. Для оценки устойчивости в правую часть уравнений (5) и (6) был добавлен гауссов шум (0,0.05). Модифицированная сеть показала рост погрешности на 18%, а метод Рунге–Кутты – на 45%.

Заключение

В работе реализована авторская модификация нейронной сети, заключающаяся в усложнении архитектуры и улучшении алгоритма обучения за счёт использования адаптивного шага обучения. Сравнение с методами Эйлера и Рунге–Кутты 4-го порядка показало, что предложенный нейросетевой подход достигает сопоставимой точности, обеспечивая решение в аналитической форме. Дополнительно исследована устойчивость к шуму. Предложенный подход рекомендуется для задач, где важна аналитическая форма решения и допустимы умеренные вычислительные ресурсы.

Список использованных источников

1. Ахметов, И.З. Решение дифференциальных уравнений с помощью нейросетей / И.З. Ахметов, А.Ф. Галимянов // Технологические инновации и научные открытия. – 2023. – С. 8–18.
2. Lagaris, I.E. Artificial neural networks for solving ordinary and partial differential equations / I.E. Lagaris, A. Likas, D.I. Fotiadis // Department of Computer Science. – 1997. – V. 9, № 5. – P. 974–1000.
3. Lee, H. Neural algorithms for solving differential equations / H. Lee, I. Kang // Journal of Computational Physics. – 1990. – V. 91. – P. 110–117.
4. Камке, Э. Справочник по обыкновенным дифференциальным уравнениям / Э. Камке. – Издательство «Наука», 1976. – 589 с.

APPLICATION OF NEURAL NETWORKS FOR SOLVING ORDINARY DIFFERENTIAL EQUATIONS

G.A. Margaryan¹

¹*Samara University, Samara
e-mail: 2020-00948@students.ssau.ru*

Abstract

The paper presents a feedforward neural network for solving ordinary differential equations of higher orders. The main focus is on modifying the network to prevent overfitting. When comparing the obtained solution with the Euler and Runge-Kutta methods, a reduction in error by a factor of 20 to 60 was observed. Additionally, the paper provides an analysis of computational costs, noise tolerance, and scalability.

Keywords: ordinary differential equations, neural network, collocation method, Euler method, error, Runge–Kutta method, regularization.