Исследование влияния вида задачи для уравнения гармонического осциллятора на решение PINN

Донецков А.Д.,* Бакакин В.Д., Жулев Е.М.
НИЯУ МИФИ

19 декабря 2024 г.

Аннотация

В данном отчёте рассматривается применение метода физически – информированных нейронных сетей **PINN** (Physics-Informed Neural Networks) для решения уравнения гармонического осциллятора с вынуждающей силой. Исследуется влияние различных формулировок задачи Коши на точность и сходимость метода PINN. Представлены сравнительные результаты, демонстрирующие различия в скорости сходимости, устойчивости обучения и точности.

1 Введение

Одной из важных задач в области вычислительной физики и механики является решение уравнений движения гармонического осциллятора. Классический гармонический осциллятор описывается различными дифференциальными уравнениями как второго, так и первого порядка. В последнее десятилетие активно развиваются нейросетевые методы решения дифференциальных уравнений, в частности метод физически-информированных нейронных сетей (PINN) [1, 2].

Цель данного исследования — сравнить влияние различных формулировок задачи Коши для уравнения гармонического осциллятора на точность и эффективность метода PINN.

2 Постановка задачи

Рассматриваются три различных формулировки задачи Коши для уравнения гармонического осциллятора с вынуждающей силой, по 2 набора параметров для каждой - для случаев резонанса и его отсутствия:

*E-mail: andrey.donetskov@gmail.com

2.1 ОДУ второго порядка

Уравнение вида

$$
\begin{cases}
\dfrac{d^2x}{dt^2} + \omega_0^2 x = -A\cos(\omega t), \\
x(0) = x_0, \\
\dfrac{dx}{dt}(0) = v_0.
\end{cases}
\tag{1}
$$

непосредственно описывает динамику гармонического осциллятора.

Применение:

- **Физика колебаний:** Моделирование механических систем, таких как пружинные маятники или электрические колебательные контуры, где присутствуют гармонические колебания.
- **Анализ устойчивости:** Исследование устойчивости систем, подверженных периодическим внешним воздействиям.

2.2 Системы ОДУ первого порядка

$$
\begin{cases}
\dfrac{dx}{dt} = y, \\
\dfrac{dy}{dt} = -\omega_0^2 x - A\cos(\omega t), \\
x(0) = x_0, \\
y(0) = v_0.
\end{cases}
\tag{2}
$$

$$
\begin{cases}
\dfrac{dx}{dt} = \omega_0 y - \dfrac{A}{\omega}\sin(\omega t), \\
\dfrac{dy}{dt} = -\omega_0 x, \\
x(0) = x_0, \\
y(0) = \dfrac{v_0}{\omega_0}.
\end{cases}
\tag{3}
$$

Применение:

- **Численные методы:** Многие численные алгоритмы, такие как методы Рунге – Кутты, разработаны для систем первого порядка, что делает эту форму удобной для компьютерного моделирования.
- **Сложные системы:** Анализ систем с несколькими степенями свободы, где каждая степень описывается своим уравнением первого порядка.

Примечание: Разные системы дифференциальных уравнений первого порядка могут быть полезны в случаях, когда форма упрощает аналитическое или численное решение, в зависимости от характера внешней силы или других факторов.

3 Методология

Для сравнительного анализа во всех 3 случаях использовалась одинаковая конфигурация нейросети: 1 входной нейрон принимает время, 5 промежуточных слоёв по 20 нейронов, один или 2 выходных нейрона. Функцией активации был выбран синус. Для решения задачи использовался метод PINN. Основная идея PINN состоит в обучении нейронной сети, которая аппроксимирует решение дифференциального уравнения. Функционал потерь $loss = loss_{interior} + \lambda_{bc} * loss_{boundary}$ состоит из двух частей:

1. Потери на уравнении (MSE по residual), учитывающие форму дифференциального уравнения.

2. Потери на граничных условиях (MSE между предсказанием и истинным значением).

Обучение модели проводилось с использованием оптимизатора, такого как Adam, для минимизации функции потерь. Этот оптимизатор показывает себя наиболее эффективно для PINN.

4 Результаты

В ходе экспериментов было получено следующее:

- **ОДУ второго порядка(1):** Обеспечивает высокую точность и быструю сходимость, но может попадать в локальные минимумы. См. рисунки 1 2.

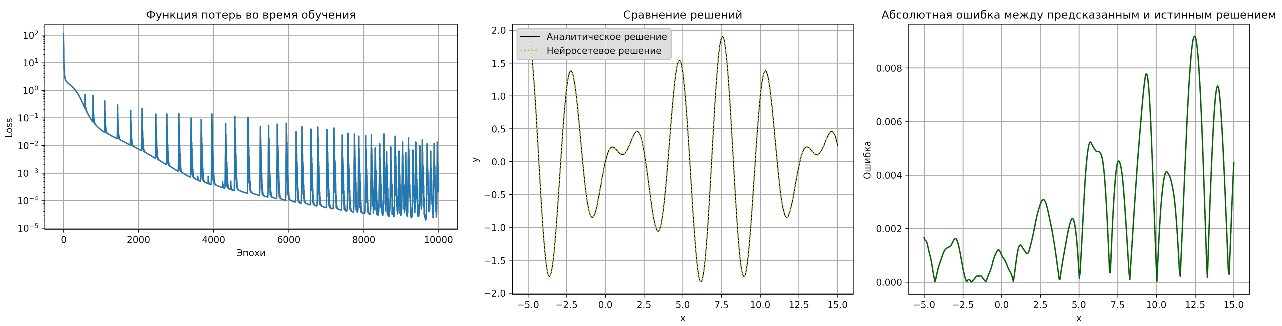

Рис. 1: Функция потерь для ОДУ второго порядка

- **Система ОДУ первого порядка(2):** Демонстрирует более стабильный, но медленный процесс обучения и менее точные результаты. См. рисунки 3 4.

- **Альтернативная система ОДУ первого порядка(3):** Достигается точность, близкая к случаю второго порядка, при этом потери более стабильны. См. рисунки 5 6.

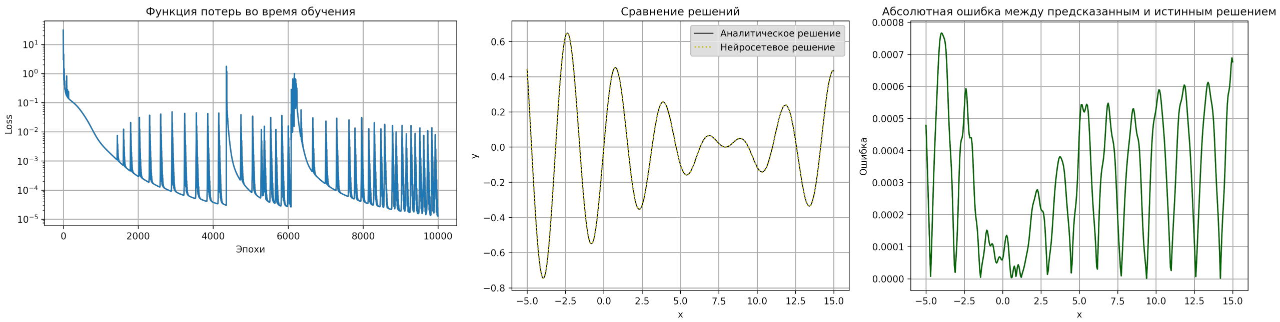

Рис. 2: Функция потерь для ОДУ второго порядка с резонансом

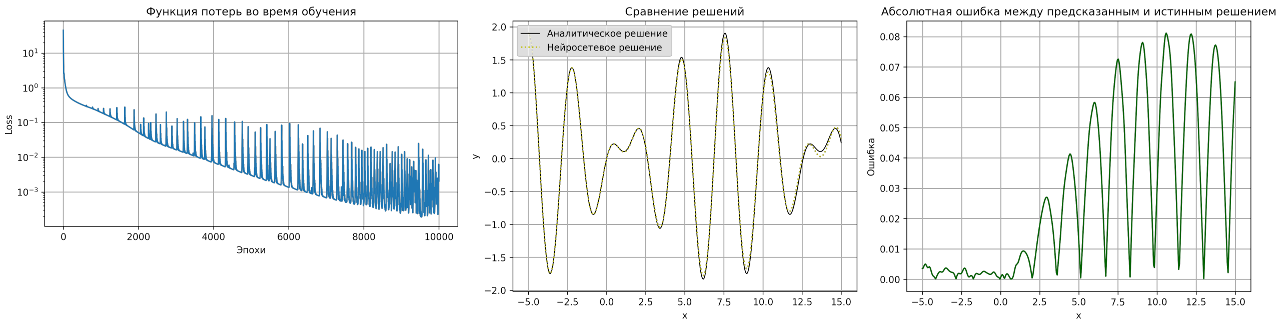

Рис. 3: Функция потерь для системы ОДУ первого порядка

Рис. 4: Функция потерь для системы ОДУ первого порядка с резонансом

Рис. 5: Функция потерь для альтернативной системы ОДУ

5 Обсуждение результатов

Из результатов видно, что выбор формы постановки задачи влияет на сходимость, время и стабильность обучения:

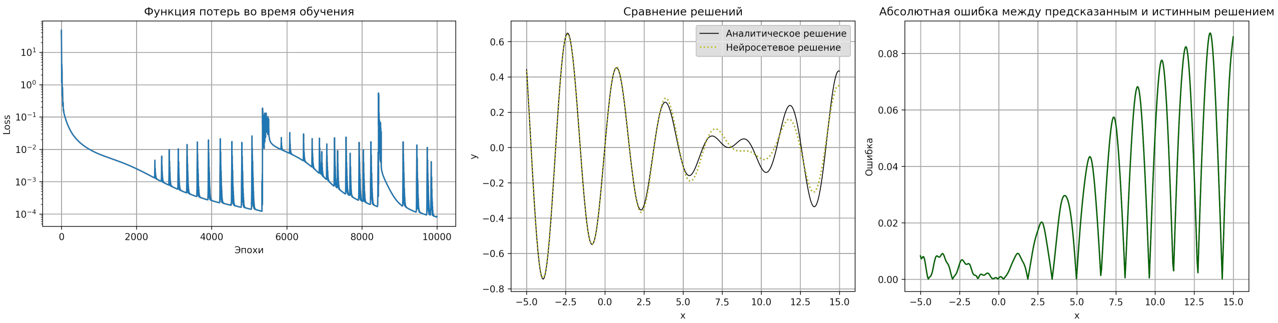

Рис. 6: Функция потерь для альтернативной системы ОДУ с резонансом

- Использование исходного уравнения второго порядка даёт быструю сходимость к точному решению, однако могут возникать проблемы с попаданием в локальные минимумы.
- Преобразование уравнения к системе первого порядка повышает устойчивость обучения, но может снижать точность и увеличивать время сходимости.
- Альтернативная система первого порядка позволяет достичь результата, близкого к исходному уравнению, при этом обладая более стабильным процессом оптимизации.

6 Выводы

Проведённое исследование показывает, что выбор постановки задачи Коши существенно влияет на эффективность и точность метода PINN. Для конкретных приложений имеет смысл сравнить различные формулировки задачи, чтобы подобрать оптимальный вариант. Полный код для воспроизведения результатов доступен в репозитории на GitHub: https://github.com/PracticalOscillations/Practice3. Мы приглашаем заинтересованных исследователей использовать и развивать представленный подход.

Список литературы

[1] Lagaris I.E., Likas A., Fotiadis D.I. Artificial neural networks for solving ordinary and partial differential equations. //IEEE Transactions on Neural Networks, 1998. Т. 9. № 5. С. 987–1000.

[2] Raissi M., Perdikaris P., Karniadakis G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. //Journal of Computational Physics, 2019. Т. 378. С. 686–707.

[3] PyTorch documentation.