

Применение физически информированного глубокого обучения для решения дифференциальных уравнений

Даниил Лаптев

12 мая 2024 г.

Содержание

1 Введение	2
1.1 Обзор литературы	2
2 Методология Physics-Informed Neural Networks	2
3 Дизайн экспериментов	4
3.1 Рассматриваемые дифференциальные уравнения	4
3.1.1 Гармонический осциллятор с затуханием	4
3.1.2 Система Лотки-Вольтерры	4
3.1.3 Уравнение диффузии	5
3.1.4 Система Лоренца	6
3.1.5 Модель Грея-Скотта	6
4 Результаты экспериментов	6
4.1 Решение дифференциальных уравнений	6
4.2	7
5 Анализ гиперпараметров и процесса обучения	7
Заключение	7
A Методы получения данных	7
B Получение численного решения дифференциальных уравнений	7
C Подбор гиперпараметров и метод анализа их роли	7
D Сбор информации о динамике обучения	7
E Анализ Фурье	7
F	7
Список источников	7

[illegible]

1 Введение

1.1 Обзор литературы

2 Методология Physics-Informed Neural Networks

Рассмотрим общую постановку проблемы. Решением ДУ называется функция u , определённая на пространстве Ω , которая удовлетворяет условиям:

$$\begin{aligned}\mathcal{D}u &= f, \\ \mathcal{B}u &= g,\end{aligned}\tag{1}$$

где \mathcal{D} - дифференциальный оператор для всего домена, \mathcal{B} - дифференциальный оператор на границе домена, f и g - какие-либо заданные функции.

Опираясь на свойство универсальной аппроксимации [2], мы можем пожелать приблизить искомую функцию некоторой нейросетевой моделью с параметрами θ . Тогда проблема решения ДУ сведётся к задаче оптимизации (обучения) относительно какой-либо функции потерь.

Пусть $\mathcal{N}(x; \theta)$, где $x \in \Omega$ - нейросетевая модель. Для удобства будем писать \mathcal{N}_θ . То, насколько нейросетевая модель удовлетворяет условиям (1), можно выразить следующим образом:

$$\begin{aligned} L_{\mathcal{D}}(\theta) &= \|\mathcal{DN}_{\theta} - f\|, \\ L_{\mathcal{B}}(\theta) &= \|\mathcal{BN}_{\theta} - g\|, \end{aligned} \quad (2)$$

Тогда оптимизируемую функцию потерь можно записать как

$$\mathcal{L}(\theta) = w_1 L_{\mathcal{B}}(\theta) + w_2 L_{\mathcal{D}}(\theta), \quad (3)$$

где коэффициенты w_1 и w_2 нужны для того, чтобы балансировать между важностью граничного и внутреннего термов функции потерь. В тех или иных условиях может потребоваться пожертвовать точностью на границе для того, чтобы улучшить точность внутри домена, и наоборот. Для удобства мы введём параметр C и определим $w_1 = C$, $w_2 = (1 - C)$, как в работе [3], тем самым уменьшив число гиперпараметров на один.

В конечном итоге для какой-либо фиксированной нейросетевой архитектуры мы ищем такой набор параметров θ^* , что

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta)$$

Рассмотрим (2) и (3). Минимизацию функции потерь $L_{\mathcal{B}}$ можно воспринимать как классическое обучение с учителем, поскольку на границе нам известны значения искомой функции и/или её производных; в свою очередь, $L_{\mathcal{D}}$ является регуляризационным термом, который ограничивает пространство допустимых параметров. L1 и L2 регуляризация направляет оптимизатор в те области, в которых норма параметров будет небольшой; регуляризация с помощью $L_{\mathcal{D}}$, который часто называется PINN-термом функции

потерь, направляет оптимизатор в те области, в которых нейросетевая модель будет удовлетворять дифференциальному уравнению внутри домена. Иными словами, мы ищем параметры нейросети в таком множестве, которое можно описать следующим образом:

$$\Theta = \{\theta : \|\mathcal{DN}_\theta - f\| \approx 0 \quad \text{и} \quad \|\mathcal{BN}_\theta - g\| \approx 0\}$$

Поскольку на практике мы не можем достичь равенства нулю, мы используем знак \approx . В разных контекстах близость к нулю может пониматься по-разному. Если решение поставленной задачи существует и единственно, то минимизация описанных выше функций потерь означает также и приближение искомой функции, причём ровно одним способом.

Чтобы получить значение функции потерь, обычно используется Монте-Карло оценка L2 нормы, по-другому известная как Mean Squared Error. Например, если на границе нами выбрано (или нам дано) N_B пар вида (x_i, g_i) , а внутри домена мы каким-либо образом выбираем N_D точек x_i , то функцию ошибки можно оценить как

$$L_D(\theta) = \frac{1}{N_D} \sum_{i=1}^{N_D} (\mathcal{DN}_\theta(x_i) - f(x_i))^2,$$

$$L_B(\theta) = \frac{1}{N_B} \sum_{i=1}^{N_B} (\mathcal{BN}_\theta(x_i) - g(x_i))^2$$

Формально разницы между этими оценками нет, однако нередко на границе заданы сами значения искомой функции (например, если граничные условия поставлены в форме Дирихле) и/или её производных (например для задачи Коши, в которой известно значение функции и её производных в начальный момент времени). Интуитивно мы можем понимать L_B как условие на точное значение искомой функции, а L_D как условие на характер поведения искомой функции на всём домене.

Дифференциальный оператор, как граничный, так и внутренний, легче всего приблизить с помощью алгоритма автоматического дифференцирования. Тогда частные производные любого порядка могут быть вычислены для любой точки, в которой мы оцениваем значение нейросетевой модели. Единственное условие - наличие функции активации подходящей гладкости, чтобы производные высших порядков вообще существовали.

Процесс оптимизации чаще всего реализуется с помощью алгоритмов Adam и L-BFGS [1], нередко в комбинации - сначала происходит тренировка с помощью Adam, затем тренировка продолжается с помощью L-BFGS. В нашем исследовании мы ограничимся алгоритмом Adam.

В качестве нейросетевой архитектуры мы используем модели прямого распространения (FF), которые представляются как композиция аффинных и нелинейных преобразований:

$$\begin{aligned} Linear_i(x) &= W_i x + b_i, \\ FF(z) &= Linear_L(\dots \phi(Linear_1(z))), \end{aligned} \tag{4}$$

где ϕ - некая функция активации.

Для наших экспериментов мы возьмём такие модели, у которых количество строк в матрицах W_1, \dots, W_{L-1} одинаково. Пусть это количество (ширина нейросети) записывается как W . Тогда модель глубины L и ширины W будет обозначаться как $FF_{L,W}$. Если каждый параметр в нейросети пронумерован и всего их p , то вектор $\theta \in \mathbb{R}^p$ можно взаимно-однозначно соотнести с параметрами нейросетевой модели.

В общем случае нейросетевые модели могут быть произвольными.

3 Дизайн экспериментов

3.1 Рассматриваемые дифференциальные уравнения

Свойство универсальной аппроксимации лишь гарантирует нам потенциальную возможность аппроксимировать искомую функцию некоторой нейросетевой моделью, однако для того, чтобы это сделать, необходимо подобрать подходящий алгоритм обучения. Последний мы понимаем как совокупность нейросетевой модели, формулировки функции потерь, а также гиперпараметров тренировочного процесса. В данном разделе мы опишем простейший способ решения некоторых ДУ с помощью PINNs, и сфокусируемся на проблемах, которые возникают в процессе обучения и выбора оптимальных гиперпараметров.

Для того, чтобы оценивать качество PINNs, мы использовали и реализовали некоторые классические численные методы для получения ground truth значений искомой функции (подробнее см. в разделе ??). Чем точнее будет получено решение ДУ, тем точнее будет оценка качества работы нейросетевой модели.

3.1.1 Гармонический осциллятор с затуханием

Система описывается следующим образом:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_0 \frac{dx}{dt} + \omega_0^2 x = 0, \quad (5)$$

где функция $x(t) : \mathbb{R} \rightarrow \mathbb{R}$ описывает положение осциллирующей массы, а параметры ζ и ω_0 влияют на характер осцилляции и её затухания.

Пусть дано начальное положение массы $x(0) = x_0$ и начальная скорость $v(0) = \frac{dx}{dt}(0) = v_0$. Зафиксируем некоторую нейросетевую архитектуру $\mathcal{N}(z; \theta)$. Тогда мы должны отыскать такие параметры θ , которые минимизируют следующий функционал:

$$L(\theta) = \frac{C}{2} \left\| \begin{bmatrix} \mathcal{N}_\theta(0) - x_0 \\ \frac{d\mathcal{N}_\theta}{dt}(0) - v_0 \end{bmatrix} \right\|_2^2 + \frac{1-C}{N_{\mathcal{D}}} \sum_{i=1}^{N_{\mathcal{D}}} \left(\left[\frac{d^2}{dt^2} + 2\zeta\omega_0 \frac{d}{dt} + \omega_0^2 \right] \mathcal{N}_\theta(t_i) \right)^2$$

3.1.2 Система Лотки-Вольтерры

Рассмотрим модель взаимодействия двух биологических видов, один из которых выполняет роль хищника, другой - жертвы. Такая система часто моделируется с помощью системы ОДУ Лотки-Вольтерры. Пусть число особей в их популяциях описывается соответственно непрерывными функциями $y(t)$ и $x(t)$. Тогда динамика числа особей будет выражена как

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta xy \\ \frac{dy}{dt} = \delta y - \gamma y \end{cases}, \quad (6)$$

где параметры $\alpha, \beta, \delta, \gamma$ характеризуют рождаемость и смертность популяций.

Пусть нейросетевая модель $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{R}^2$ будет аппроксимировать значение сразу двух искомых функций в момент времени t посредством двумерного вектора на её выходе:

$$\mathcal{N}(t; \theta) = \begin{bmatrix} \mathcal{X}(t; \theta) \\ \mathcal{Y}(t; \theta) \end{bmatrix} \approx \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

Пусть даны начальные значения x_0, y_0 . Нам необходимо найти такие параметры θ ,

которые минимизируют следующие функции ошибки:

$$\begin{aligned}\mathcal{L}_{\mathcal{I}} &= \frac{1}{2} \left\| \begin{bmatrix} \mathcal{X}(0) - x(0) \\ \mathcal{Y}(0) - y(0) \end{bmatrix} \right\|_2^2 \\ \mathcal{L}_{\mathcal{X}} &= \frac{1}{N_{\mathcal{D}}} \sum_{i=1}^{N_{\mathcal{D}}} \left(\frac{d\mathcal{X}}{dt}(t_i) - \alpha\mathcal{X}(t_i) + \beta\mathcal{X}(t_i)\mathcal{Y}(t_i) \right)^2 \\ \mathcal{L}_{\mathcal{Y}} &= \frac{1}{N_{\mathcal{D}}} \sum_{i=1}^{N_{\mathcal{D}}} \left(\frac{d\mathcal{Y}}{dt}(t_i) - \delta\mathcal{X}(t_i)\mathcal{Y}(t_i) + \gamma\mathcal{Y}(t_i) \right)^2\end{aligned}\tag{7}$$

В качестве итоговой функции потерь мы возьмём их линейную комбинацию:

$$L_{\mathcal{T}} = C L_{\mathcal{I}} + (1 - C)(L_{\mathcal{X}} + L_{\mathcal{Y}})$$

3.1.3 Уравнение диффузии

Эта модель используется как самостоятельное описание процесса диффузии или как особый случай некоторых других моделей. Мы рассмотрим задачу Дирихле для одномерного случая.

Пусть имеется функция $u(\mathbf{x}, t) : \mathbb{R}^{N+1} \rightarrow \mathbb{R}$, которая описывает, например, концентрацию вещества в какой-либо точке пространства и времени (в нашем случае $N = 1$). Пусть число D , называемое коэффициентом диффузии, является постоянным на всей ограниченной области $[A, B] \times [0, T]$. Кроме того, на границах этой области нам известно поведение искомой функции. Тогда краевую задачу для уравнения диффузии мы запишем как

$$\begin{aligned}\frac{\partial u}{\partial t} &= D \frac{\partial^2 u}{\partial x^2}, \\ u(x, 0) &= f(x), \\ u(A, t) &= g_1(t), \\ u(B, t) &= g_2(t),\end{aligned}\tag{8}$$

причём вместо явно заданных функций на границах мы можем иметь лишь некоторый набор измерений. Методы машинного обучения позволят приблизить решение на основе неполных данных, что является ещё одним достоинством PINNs.

Пусть при $t = 0$ и на границах выбирается соответственно N_I , N_A , N_B точек вместе с заданными в них значениями искомой функции. Внутри области мы берём N_D произвольных точек. Тогда функции потерь мы можем записать как

$$\begin{aligned}L_{\mathcal{I}} &= \frac{1}{N_I} \sum_{i=1}^{N_I} (\mathcal{N}(x_i, 0) - f(x_i))^2, \\ L_B &= \frac{1}{N_A} \sum_{i=1}^{N_A} (\mathcal{N}(A, t_i) - g_1(t_i))^2 + \frac{1}{N_B} \sum_{i=1}^{N_B} (\mathcal{N}(B, t_i) - g_2(t_i))^2, \\ L_{\mathcal{D}} &= \frac{1}{N_D} \sum_{i=1}^{N_D} \left(\frac{\partial \mathcal{N}}{\partial t}(x_i, t_i) - D \frac{\partial^2 \mathcal{N}}{\partial x^2}(x_i, t_i) \right)^2\end{aligned}\tag{9}$$

Полная функция потерь в нашей реализации выглядит следующим образом:

$$L = C(L_{\mathcal{I}} + L_B) + (1 - C)L_{\mathcal{D}}$$

3.1.4 Система Лоренца

3.1.5 Модель Грея-Скотта

4 Результаты экспериментов

4.1 Решение дифференциальных уравнений

Подводя итоги, мы можем охарактеризовать основные проблемы обучения PINNs, которые нам удалось выявить в результате экспериментов:

1. Нейросетевые модели очень чувствительны к инициализации весов. Эксперименты показали, что от неё в первую очередь зависит успех или неуспех обучения. Для разных задач и разных нейросетевых моделей подходящая инициализация оказалась также различной.
2. Динамика искомой функции значительно влияет на то, как нейросетевая модель обучается и какую информацию она извлекает из функции потерь. Кроме того, выбор функции активации также влияет на ход обучения из-за эффекта затухающего градиента и сатурации (свойства, при котором значение функции активации ограничено асимптотически сверху, как для \tanh или сигмоиды), которые мешают нейросетевой модели извлекать информацию при больших значениях входного аргумента.
3. В силу особенностей формулировки задачи, неясно, что считать переобучением нейросетевой модели. Вместо переобучения, которое обычно выражается в ухудшении качества работы модели после какого-то этапа обучения, мы, наоборот, сталкиваемся с трудностями получения всё более низкого значения функции потерь. Качество может ухудшиться, но только вместе с увеличением значения функции потерь на «тренировочном» наборе данных (граничные условия + точки коллокации). Это подтверждает наше предположение о том, что если задача поставлена корректно, то нейросетевая модель сойдётся к её решению.
4. В некоторых случаях ухудшение качества работы нейросетевой модели выражалось в том, что в один момент она как бы сбрасывалась на более ранний этап обучения, принимала такую форму, которую уже имела раньше. Однако во многих случаях после этого обучение продолжалось как будто бы заново, и мы могли достигать всё более низких значений функции потерь и ошибок, притом, что нормы каждого слоя нейросетевой модели практически монотонно изменялись. Этот эффект мы называли двойным обучением.
5. Пронаблюдав за динамикой обучения моделей на задачи гармонического осциллятора и системы Лотки-Вольтерры, мы выявили эффект, который мы характеризуем как обучение слева направо. Нейросетевая модель сначала повторяет форму искомой функции на левой части домена, затем на правой части домена. Это скорее всего связано с наблюдениями 2 и 4, т.е. с затухающими градиентами и сатурацией, а также с тем, что нейросетевая модель знает значение функции лишь в нуле, а всю остальную динамику ей приходится учить, опираясь на эту информацию (и таким образом можно сказать, что нейросеть аппроксимирует информацию справа, основываясь на информации слева, используя выученную динамику слева как опору для дальнейшего обучения).

5 Анализ гиперпараметров и процесса обучения

Заключение

А Методы получения данных

В Получение численного решения дифференциальных уравнений

С Подбор гиперпараметров и метод анализа их роли

Д Сбор информации о динамике обучения

Е Анализ Фурье

Ф

Список источников

- [1] Salvatore Cuomo и др. «Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What's Next». В: *Journal of Scientific Computing* 92.3 (июль 2022), с. 88. ISSN: 1573-7691. DOI: [10.1007/s10915-022-01939-z](https://doi.org/10.1007/s10915-022-01939-z). URL: <https://doi.org/10.1007/s10915-022-01939-z>.
- [2] Anastasis Kratsios. «The Universal Approximation Property». В: *Annals of Mathematics and Artificial Intelligence* 89.5 (июнь 2021), с. 435—469. ISSN: 1573-7470. DOI: [10.1007/s10472-020-09723-1](https://doi.org/10.1007/s10472-020-09723-1). URL: <https://doi.org/10.1007/s10472-020-09723-1>.
- [3] Remco van der Meer, Cornelis Oosterlee и Anastasia Borovykh. *Optimally weighted loss functions for solving PDEs with Neural Networks*. 2021. arXiv: [2002.06269](https://arxiv.org/abs/2002.06269) [math.NA].