



¹ВШЭ, ²Сколтех, ³ВК

26/10/2025

Физически-информированные нейронные сети в инженерных задачах

Кирилл Кацуба ^{1, 2}, Олег Руденко ¹, **Александр Тараканов** ^{1, 3}



Содержание

Введение и мотивация

Физически-информированные нейронные сети (PINNs)

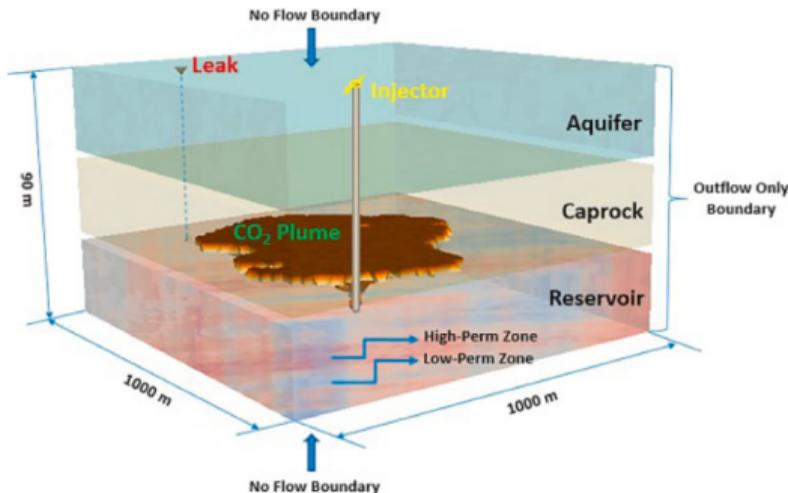
Примеры

Заключительные замечания



Примеры вычислительно сложной задачи

Пример задачи - закачка углекислого газа на долгосрочное хранение.



Закачивается определенный объем углекислого газа → оценивается объем утечки.

Идеальный результат - распределение массы утечки относительно неизвестных параметров.



Способы решения

Самый простой способ - Монте-Карло моделирование.

Проблемы для сложных моделей, где один расчет занимает несколько часов.

На практике получается собрать датасет примерно из 1000 точек.

Можно построить прокси-модель, которую можно использовать для Монте-Карло оценок.

Исторически использовались методы, неплохо работающие на небольших датасетах: SVM, полиномы, решающие деревья.

Новый тренд - нейросети в духе Физически Информированных Нейронных Сетей (PINNs)



Содержание

Введение и мотивация

Физически-информированные нейронные сети (PINNs)

Примеры

Заключительные замечания



PINNs на примере уравнения Пуассона

Уравнение Пуассона (мембрана в поле тяжести), $x \in G$:

$$-\Delta u(x) = f(x)$$

Границные условия:

$$u|_{\partial G} = 0$$

Этапы построение приближения точного решения $u(x)$ при помощи PINNs $\mathcal{N}(x)$:

- генерация сэмплов из области G и ее границы ∂G
- Монте-Карло оценка функции потерь для граничных условий: $\mathcal{L}_{IBC} \approx \sum_1^N \frac{1}{N} (\mathcal{N}(x_i))^2$
- Монте-Карло оценка функции потерь для дифференциального уравнения:
$$\mathcal{L}_{PDE} \approx \sum_1^N \frac{1}{N} (\Delta \mathcal{N}(x_i) + f(x_i))^2$$
- вычисление функций потерь для тех точек, в которых известно точное решение (их может и не быть): $\mathcal{L}_{SIM} \approx \sum_1^N \frac{1}{N} (\mathcal{N}(x_i^*) - u(x_i^*))^2$
- взвешивание функций ошибок: $\mathcal{L} = w_{SIM} \mathcal{L}_{SIM} + w_{IBC} \mathcal{L}_{IBC} + w_{PDE} \mathcal{L}_{PDE}$
- оптимизация градиентным спуском



Достоинства и недостатки и основные направления исследований

Достоинства метода:

- прост в понимании и обобщении
- легко программировать
- почти единственный метод для многомерных УрЧП (уравнение Шредингера)

Недостатки:

- сложно контролировать точность метода
- не всегда получается гарантировать выполнение ограничений, например, законов сохранения

Основные направления исследований:

- методы сэмплирования точек для проверки выполнения граничных условий и дифференциального уравнения
- методы взешивания функций потерь



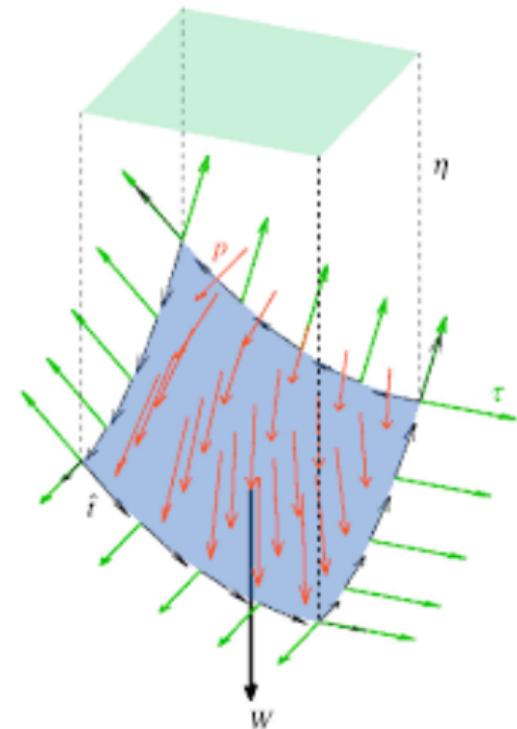
Наш вклад

Новый алгоритм взвешивания функций ошибок:

- задаются целевые значения для каждой функции ошибок:
 κ_{SIM} , κ_{IBC} , κ_{PDE}
- нормировка функций потерь $l_{\text{SIM}} = \mathcal{L}_{\text{SIM}} / \kappa_{\text{SIM}}$,
 $l_{\text{IBC}} = \mathcal{L}_{\text{IBC}} / \kappa_{\text{IBC}}$, $l_{\text{PDE}} = \mathcal{L}_{\text{PDE}} / \kappa_{\text{PDE}}$
- иерархия функций ошибок за счет соответствующих весов:

$$\left\{ \begin{array}{l} w_{\text{SIM}} \propto 1, \\ w_{\text{IBC}} \propto \exp(-l_{\text{SIM}}), \\ w_{\text{PDE}} \propto \exp(-\max(l_{\text{SIM}}, l_{\text{IBC}})), \end{array} \right.$$

Новая ниша для применения PINNs: мало данных и высокая нелинейность





Содержание

Введение и мотивация

Физически-информированные нейронные сети (PINNs)

Примеры

Заключительные замечания



Уравнение теплопроводности

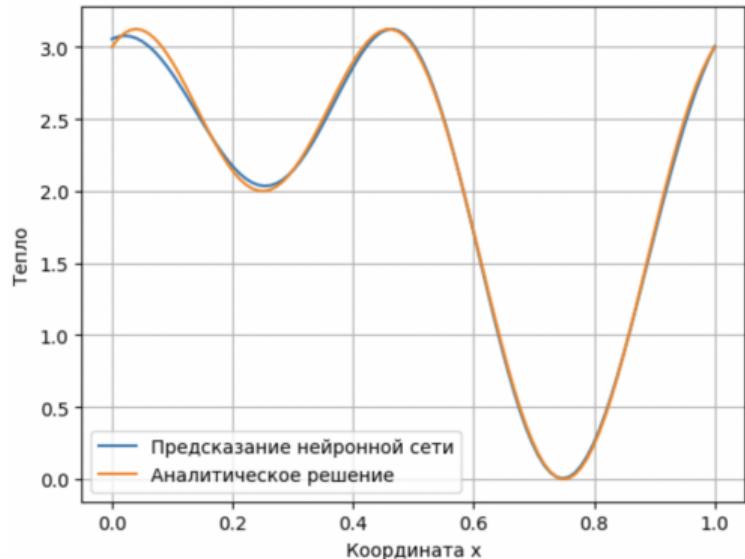
Тест на примере, в котором есть точное решение:

$$\begin{cases} \frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial x^2} = 0, \\ T(0, x) = 2 + \sin(2\pi x) + \cos(4\pi x), \\ T(t, 0) = T(t, 1). \end{cases}$$

Точное решение:

$$T(t, x) = 2 + e^{-4\pi^2 t} \sin(2\pi x) + e^{-16\pi^2 t} \cos(4\pi x)$$

Нейросеть обучалась "без подсказок" для вычисления решения при $x \in [0, 1]$, $t \in [0; 0.1]$ с периодическими граничными условиями.





Гидродинамика в пористой среде

Синтетический пример двухфазного вытеснения двух несжимаемых жидкостей в духе одного из классических способов нефтедобычи.

Две жидкости (вода w и нефть o) занимают некоторые доли объема в пористой среде s_w и s_o и движутся согласно законам сохранения:

$$\begin{cases} \phi \frac{\partial s_w}{\partial t} + \nabla \cdot (u_w) = 0, \\ \phi \frac{\partial s_o}{\partial t} + \nabla \cdot (u_o) = 0. \end{cases}$$

ϕ - пористость - доля объема пористой среды, в котором могут двигаться жидкости. Локальное свойство среды.

Жидкости движутся за счет перепада давлений:

$$\begin{cases} u_w = -\frac{k k_w(s_w, s_o)}{\mu_w} \nabla P, \\ u_o = -\frac{k k_o(s_w, s_o)}{\mu_o} \nabla P \end{cases}$$

Здесь k - проницаемость (локальное свойство среды), P - давление, k_w, k_o - относительные фазовые проницаемости, заданные в параметрическом виде:

$$\begin{cases} k_w = k_{\max, w} s_w^{q_w}, \\ k_o = k_{\max, o} s_o^{q_o}. \end{cases}$$



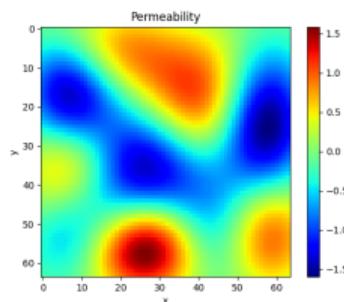
Начальные и граничные условия

В начальный момент времени все заполнено только нефтью $s_o = 1.0$, $s_w = 0.0$

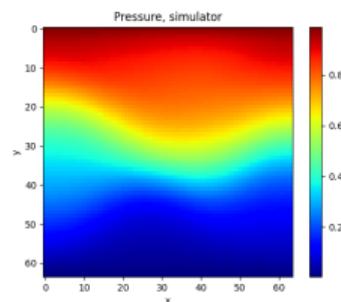
Давление на верхней стороне квадрата всегда 1, на нижней - 0

Боковые стороны непроницаемы (зануляется нормальная компонента скорости)

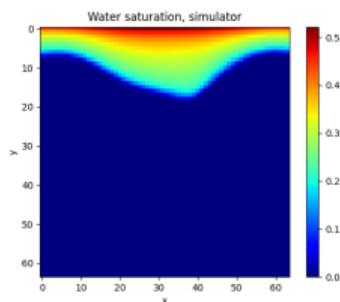
На верхней стороне квадрата насыщенность воды 1, нефти 0 в любой момент времени (закачка воды)



(a) Проницаемость



(b) Давление



(c) Насыщенность



Постановка задачи

Обучить прокси-модель для аппроксимации нысыщенности в любой точке и в любой момент времени из заданного интервала, при различных параметрах кривых относительных фазовых проницаемостей.

Нелинейностью системы можно управлять, меняя диапазон, в котором могут меняться параметры модели (чем больше диапазон, тем сложнее поведение).

Протестировать поведение разработанного алгоритма в различных сценариях сложности (нелинейности) системы.

Для этого фиксировался датасет из 500 точек и сравнивались две модели: XGBoost, PINNs.



Случай слабой нелинейности

Metric	Satur. train	Satur. test	Press. train	Press. test
MSE	$3.1 \cdot 10^{-5}$	$2.25 \cdot 10^{-4}$	$2.13 \cdot 10^{-4}$	$1.50 \cdot 10^{-4}$
MAE	$8.39 \cdot 10^{-4}$	$3.79 \cdot 10^{-3}$	$4.48 \cdot 10^{-3}$	$2.43 \cdot 10^{-3}$
R^2	0.996513	0.975380	0.998070	0.998360
RMSE	$5.565 \cdot 10^{-3}$	$1.4989 \cdot 10^{-2}$	$1.4590 \cdot 10^{-2}$	$1.2246 \cdot 10^{-2}$

Таблица: Случай слабой нелинейности, PINNs

Metric	Satur. train	Satur. test	Press. train	Press. test
MSE	$3.00 \cdot 10^{-7}$	$1.657 \cdot 10^{-3}$	$2.28 \cdot 10^{-4}$	$2.28 \cdot 10^{-4}$
MAE	$9.31 \cdot 10^{-12}$	$1.4044 \cdot 10^{-2}$	$1.0457 \cdot 10^{-2}$	$1.0457 \cdot 10^{-2}$
R^2	0.9999654	0.895641	0.997869	0.997869
RMSE	$5.48 \cdot 10^{-4}$	$4.0708 \cdot 10^{-2}$	$1.5085 \cdot 10^{-2}$	$1.5085 \cdot 10^{-2}$

Таблица: Случай слабой нелинейности, XGBoost



Случай сильной нелинейности

Обе модели стали работать хуже, но точность PINNs упала не так сильно, как у XGBoost.

Metric	Water saturation		Pressure	
	PINN	XGBoost	PINN	XGBoost
MSE	$1.52 \cdot 10^{-2}$	$6.50 \cdot 10^{-1}$	$3.41 \cdot 10^{-1}$	$3.06 \cdot 10^{-1}$
MAE	$5.08 \cdot 10^{-2}$	$7.97 \cdot 10^{-1}$	$4.67 \cdot 10^{-1}$	$4.40 \cdot 10^{-1}$
RMSE	$1.23 \cdot 10^{-1}$	$8.06 \cdot 10^{-1}$	$5.84 \cdot 10^{-1}$	$5.53 \cdot 10^{-1}$

Таблица: PINNs vs XGBoost.



Содержание

Введение и мотивация

Физически-информированные нейронные сети (PINNs)

Примеры

Заключительные замечания



Выводы и планы на будущее

Что получилось увидеть:

- PINNs может аппроксимировать достаточно сложные дифференциальные уравнения
- регуляризация в виде дифференциального уравнения и граничных условий - безлимит на данные
- PINNs более устойчив к росту нелинейности по сравнению с XGBoost

Планы на будущее:

- сравнение с другими методами классического машинного обучения
- более сложные примеры