

# Применение машинного обучения

## Вычислительные методы (3)

Лю Шисян (刘世贤)

Ассистент  
Кафедра теплофизики (Э6)  
МГТУ им. Н.Э. Баумана  
[lyu@bmstu.ru](mailto:lyu@bmstu.ru)



- ① Прогноз физических свойств
- ② Физически-информированные нейронные сети
- ③ Машинно-обученный потенциал

# Содержание

- 1 Прогноз физических свойств
- 2 Физически-информированные нейронные сети
- 3 Машинно-обученный потенциал

## Зачем прогнозировать свойства нейросетями?

\* Во многих задачах физики и материаловедения **получить данные дорого**:

- эксперименты занимают недели и месяцы;
- моделирования требуют больших вычислительных ресурсов.

★ Идея: один раз посчитать / измерить ограниченный набор образцов, а затем **научить нейросеть** предсказывать свойства для новых конфигураций.

Примеры прогнозируемых величин:

- теплопроводность наноструктур;
- теплоёмкость и фононные свойства;
- прочностные и электронные характеристики материалов.

Таким образом, нейросеть выступает как **быстрый суррогат** сложных физических расчётов.

## Типовой workflow: данные → модель → прогноз

### Шаг 1. Сбор данных

- ★ Экспериментальные измерения или результаты моделирования.
- ★ Формирование таблицы «входы → выход» (например: геометрия, температура → теплопроводность).

### Шаг 2. Обучение нейросети

- ★ Делим данные на **train / test**.
- ★ Выбираем архитектуру (несколько скрытых слоёв, активация).
- ★ Обучаем сеть минимизировать функцию потерь (MSE).

### Шаг 3. Прогноз для новых условий

- ★ На вход подаём новые параметры структуры / режима.
- ★ Нейросеть мгновенно выдаёт оценку свойства вместо долгих расчётов.

# Примеры задач в теплопереносе и материаловедении

## 1. Теплопроводность наноструктур

- входы: толщина плёнки, шероховатость границы, температура;
- выход: эффективная теплопроводность  $\kappa$ ;
- применение: быстрый скрининг геометрий перед дорогими расчётами.

## 2. Материаловедение

- предсказание теплопроводности по химическому составу и структуре;
  - оценка устойчивости и функциональных свойств новых материалов.
- ★ Нейросети служат инструментом предварительного скрининга, значительно снижая потребность в ресурсоёмких расчётах, которые затем выполняются лишь для наиболее перспективных образцов.

Код реализован в Jupyter Notebook: [prediction.ipynb](#)

# Преимущества и ограничения нейросетевых моделей

## Преимущества:

- ✓ Быстрый прогноз после обучения (миллисекунды вместо часов/дней).
- ✓ Способность аппроксимировать сложные нелинейные зависимости.
- ✓ Возможность использовать разнородные данные (эксперименты + расчёты).

## Ограничения:

- ✗ Качество прогноза сильно зависит от **качества и объёма данных**.
- ✗ Плохо экстраполируют «далеко» за областью обучающих примеров.
- ✗ Часто воспринимаются как «чёрный ящик», требуется интерпретация.

## Содержание

- 1 Прогноз физических свойств
- 2 Физически-информированные нейронные сети
- 3 Машинно-обученный потенциал

# Почему нужны физически-информированные нейросети?

**Классические нейросети** обычно обучаются только на данных:

- есть набор входов и выходов;
- сеть подбирает параметры, чтобы как можно лучше аппроксимировать эти пары;
- физические законы при этом явно не учитываются.

В то же время:

- многие инженерные задачи описываются **уравнениями в частных производных** (PDE);
- экспериментальные данные могут быть **дорогими, редкими или шумными**;
- зато физические уравнения известны и несут в себе **богатую априорную информацию**.

Идея PINN как раз в том, чтобы **объединить**:

- силу нейросетей по аппроксимации сложных функций;
- и строгость физических законов, задаваемых PDE и граничными условиями.

# Что такое PINN (Physics-Informed Neural Networks)

**Физически-информированные нейронные сети (PINN) –** это класс нейросетей, которые при обучении учитывают не только **данные**, но и **физические законы** системы.

PINN объединяют два ключевых источника информации:

- ★ **Экспериментальные или численные данные:** измерения, наблюдения, начальные и граничные значения.
- ★ **Физические уравнения:** чаще всего уравнения в частных производных, например
  - ➡ уравнение теплопроводности;
  - ➡ уравнения Навье–Стокса;
  - ➡ уравнение волны и др.

## Как это работает:

- ★ невязка по PDE и граничным/начальным условиям включается прямо в **функцию потерь**;
- ★ сеть обучается так, чтобы *одновременно* согласоваться с данными и удовлетворять физике.

**Типичные области применения:** гидродинамика, теплоперенос, электромагнетизм, волновые процессы и др.

# Уравнение теплопроводности (1D)

- ★ Уравнение теплопроводности описывает **диффузию тепла** во времени по одной пространственной координате.
- ★ Это **уравнение в частных производных**, содержащее временную и пространственную производные.

## Одномерная форма

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (1)$$

- $T = T(x, t)$  —температура;
- $x$  —координата вдоль стержня (или пластины);
- $t$  —время;
- $\alpha$  —коэффициент температуропроводности.

# Решение уравнения (1D) методом конечных разностей

**Метод конечных разностей** позволяет численно решать уравнение

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2},$$

заменяя производные конечными разностями на сетке по  $x$  и  $t$ .

**Аппроксимации производных:**

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{k+1} - T_i^k}{\Delta t}, \quad \frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2}. \quad (2)$$

**Подставляя** эти аппроксимации в исходное уравнение, получаем:

$$\frac{T_i^{k+1} - T_i^k}{\Delta t} = \alpha \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2}. \quad (3)$$

## Явная конечно-разностная схема (1D)

Перепишем полученное уравнение в явном виде:

$$T_i^{k+1} = T_i^k + \gamma \left( T_{i+1}^k - 2T_i^k + T_{i-1}^k \right), \quad (4)$$

где

$$\gamma = \alpha \frac{\Delta t}{\Delta x^2}.$$

Для явной схемы справедливо условие устойчивости:

$$\gamma \leq \frac{1}{2}.$$

### ★ Пример постановки задачи:

- **границные условия:**

$$T(t, 0) = 0, \quad T(t, L) = 1;$$

- **начальное условие:**

$$T(0, x) = 0, \quad 0 < x < L.$$

Код реализован в Jupyter Notebook: [prediction.ipynb](#)

## Решение уравнения с PINN (1D-теплопроводность)

**PINN** позволяет не задавать конечно-разностную схему явно. Вместо этого при обучении нейросети минимизируется **функция потерь**, которая включает невязки по:

- ★ самому уравнению (PDE);
- ★ граничным условиям (BC);
- ★ начальному условию (IC).

**Уравнение теплопроводности (1D):**

$$T_t - \alpha T_{xx} = 0. \quad (5)$$

**Компоненты функции потерь:**

$$\mathcal{L}_{\text{eq}} = \frac{1}{N} \sum_{i=1}^N \|T_t^i - \alpha T_{xx}^i\|^2, \quad (6)$$

$$\mathcal{L}_{\text{bc}} = \frac{1}{N} \sum_{i=1}^N \|T(t, x^b) - T_{\text{bc}}\|^2, \quad \mathcal{L}_{\text{ic}} = \frac{1}{N} \sum_{i=1}^N \|T(0, x) - T_{\text{ic}}\|^2. \quad (7)$$

**Итоговая функция потерь:**

$$\boxed{\mathcal{L}_{\text{pde}} = \mathcal{L}_{\text{eq}} + \mathcal{L}_{\text{bc}} + \mathcal{L}_{\text{ic}}}. \quad (8)$$

## Содержание

- 1 Прогноз физических свойств
- 2 Физически-информированные нейронные сети
- 3 Машинно-обученный потенциал

# Потенциальные функции

## Что такое потенциальная функция?

- ★ многомерная функция, описывающая потенциальную энергию атомной системы;
- ★ вход: координаты всех атомов и их типы (элементы);
- ★ выход: скалярная величина — энергия системы.

## Квантово-механические подходы

- ➡ теория функционала плотности (DFT);
- ➡ полуэмпирический метод сильной связи (tight-binding);
- ➡ вычислительная сложность обычно порядка  $O(N^3)$ .

## Классические подходы

- ➡ аналитические выражения, не требуют решения уравнений квантовой механики;
- ➡ вычислительная сложность обычно порядка  $O(M)$ ;
- ➡ ранние подходы: эмпирические потенциалы;
- ➡ современные подходы: машинно-обученные потенциалы.

# Эмпирические vs машинно-обученные потенциалы

## Сходства

- ➡ оба относятся к классическим (не квантово-механическим) потенциалам;
- ➡ представляют собой аналитические функции и соблюдают базовые симметрии;
- ➡ вычислительная стоимость значительно ниже, чем у квантовых методов.

## Различия

- ➡ эмпирические потенциалы подгоняются по физическим величинам; машинно-обученные —тренируются сразу на энергиях и силах;
- ➡ форма эмпирических потенциалов обычно имеет явный физический смысл; у машинно-обученных потенциалов — нет;
- ➡ число параметров у эмпирических потенциалов невелико; у машинно-обученных потенциалов —обычно значительно больше.

## Классификация машинно-обученных потенциалов

### Четыре поколения по Behler

- ★ **1-е поколение:** энергия всей молекулы/системы;
- ★ **2-е поколение:** атомные энергии (по атомам);
- ★ **3-е поколение:** энергия + заряды;
- ★ **4-е поколение:** энергия + заряды + баланс зарядов.

### По архитектуре модели

- ➡ **Локальные атомные дескрипторы:**  
BPNN (2007), GAP (2010), SNAP (2015), MTP (2016),  
DP/DeepMD (2018), ACE (2019), NEP (2021)
- ➡ **Графовые сети и передача сообщений:**  
SchNet (2018)
- ➡ **Эквивариантные дескрипторы и передача сообщений:**  
NequiP (2022), MACE (2022), GACE (2024), CACE (2024),  
HotPP (2024)

## NEP: Простая архитектура сети

- ★ Входы: дескрипторы окружения атома  $i$ :  $q_\nu^{(i)}$ ,  $\nu = 1, \dots, N_{\text{des}}$ .
- ★ Скрытые нейроны:

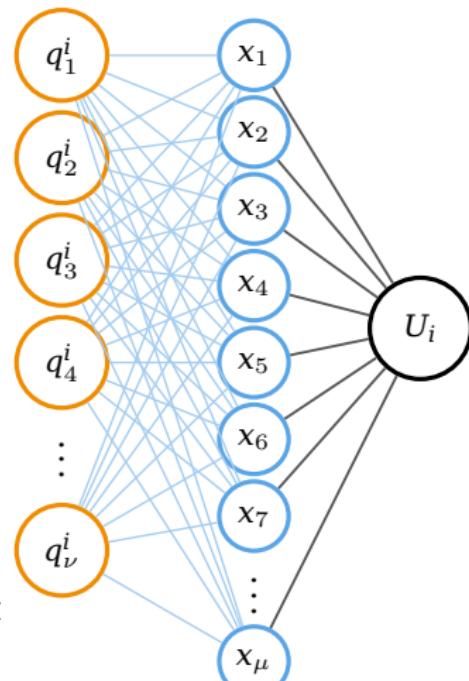
$$x_\mu = \tanh \left( \sum_{\nu=1}^{N_{\text{des}}} w_{\mu\nu}^{(0)} q_\nu^{(i)} - b_\mu^{(0)} \right),$$

•  $\mu = 1, \dots, N_{\text{neu}}$ .

- ★ Выход (атомная энергия):

$$U_i = \sum_{\mu=1}^{N_{\text{neu}}} w_\mu^{(1)} x_\mu - b^{(1)}.$$

- ★ Энергия системы:  $E = \sum_i U_i$ ; силы  $\mathbf{F} = -\partial E / \partial \mathbf{R}$ .



*Простая архитектура: один скрытый слой и активация tanh.*

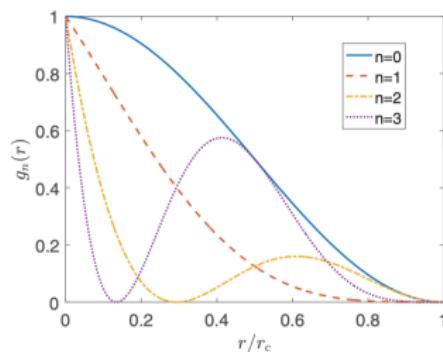
## НЕР: Локальные дескрипторы

★ Радиальные дескрипторы ( $n \geq 0$ )

$$q_n^i = \sum_{j \neq i} g_n(r_{ij}),$$

$$g_n(r_{ij}) = \frac{T_n(x) + 1}{2} f_c(r_{ij}),$$

где  $T_n$  — полиномы Чебышёва 1-го рода, а  $x \equiv 2\left(\frac{r_{ij}}{r_c} - 1\right)^2 - 1 \in [-1, 1]$ .



Функция отсечки:  $f_c(r) = \begin{cases} \frac{1}{2}\left(1 + \cos\frac{\pi r}{r_c}\right), & r \leq r_c, \\ 0, & r > r_c. \end{cases}$

★ Угловые дескрипторы ( $n \geq 0, l \geq 1$ )

$$q_{nl}^i = \sum_{j \neq i} \sum_{k \neq i} g_n(r_{ij}) g_n(r_{ik}) P_l(\cos \theta_{ijk}),$$

где  $P_l$  — полином Лежандра порядка  $l$ ,  $\theta_{ijk}$  — угол между связями  $ij$  и  $ik$ .

# NEP: функция потерь и регуляризация

## Итоговая функция потерь

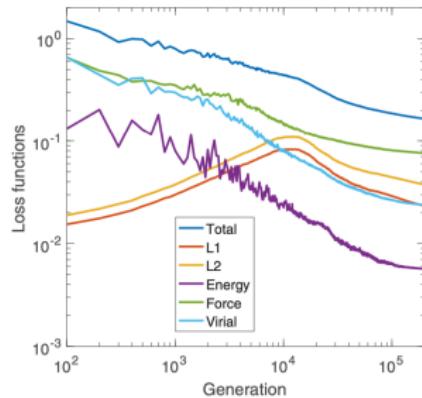
$$L(\mathbf{z}) = \lambda_e L_e(\mathbf{z}) + \lambda_f L_f(\mathbf{z}) + \lambda_v L_v(\mathbf{z}) + \lambda_1 L_1(\mathbf{z}) + \lambda_2 L_2(\mathbf{z}).$$

## Составляющие (RMSE)

$$L_e(\mathbf{z}) = \left( \frac{1}{N_{\text{str}}} \sum_{n=1}^{N_{\text{str}}} (U^{\text{NEP}}(n, \mathbf{z}) - U^{\text{tar}}(n))^2 \right)^{1/2},$$

$$L_f(\mathbf{z}) = \left( \frac{1}{3N} \sum_{i=1}^{3N} (F_i^{\text{NEP}}(\mathbf{z}) - F_i^{\text{tar}})^2 \right)^{1/2},$$

$$L_v(\mathbf{z}) = \left( \frac{1}{6N_{\text{str}}} \sum_{n=1}^{N_{\text{str}}} \sum_{\mu\nu} (W_{\mu\nu}^{\text{NEP}}(n, \mathbf{z}) - W_{\mu\nu}^{\text{tar}}(n))^2 \right)^{1/2}$$



## Регуляризация параметров

$$L_1(\mathbf{z}) = \frac{1}{N_{\text{par}}} \sum_{n=1}^{N_{\text{par}}} |z_n|, \quad L_2(\mathbf{z}) = \left( \frac{1}{N_{\text{par}}} \sum_{n=1}^{N_{\text{par}}} z_n^2 \right)^{1/2}.$$

Методы и примеры доступны на сайте [calorine](#).