

Дано:

- уравнение в области  $(x, y, t) \in (0, \pi) \times (-\pi, \pi) \times (0, 1]$ , начальное и граничные условия, а также набор точек  $(x, y, 1)$  со значениями  $u(x, y, 1)$  в них (замеры функции в некоторых точках области в самый последний момент времени).

Требуется:

- Найти неизвестную константу  $\lambda$ .
- Прислать значение  $\lambda$  и Jupyter notebook с кодом обучения и графиками обучения для как минимум 3 разных начальных приближений параметра  $\lambda$ , которые сходятся к одному значению. Telegram: [@yakefu](https://t.me/yakefu)

Советы:

- Можно взять 1D-пример [https://deepxde.readthedocs.io/en/latest/demos/pinn\\_inverse/diffusion.1d.inverse.html](https://deepxde.readthedocs.io/en/latest/demos/pinn_inverse/diffusion.1d.inverse.html) и модифицировать его код.
- Загружать данные надо через `pumpy.load()`. Ключи параметров: 'хyt' и 'u'.
- Обучение простой модели идёт долго.  
Можно начать с  $lr = 5 \cdot 10^{-4}$  и 20 тыс. итераций, потом уменьшить  $lr$  и сделать ещё итерации. Можно попробовать в конце взять  $lr = 10^{-5}$  и поставить на ~100 тыс. итераций.  
Можно попробовать сначала сделать около 3 тыс. итераций с  $lr = 10^{-3}$ , потом 50 тыс. итераций с  $lr = 10^{-4}$ , потом 100 тыс. итераций с  $lr = 10^{-5}$ . Но нужно следить за графиком функции потерь на тестовой выборке, потому что  $lr = 10^{-3}$  может оказаться слишком большим и повести обучение не в ту сторону.
- Пробуйте разные начальные приближения: больше нуля, меньше нуля.
- В граничных условиях указаны значения обычных производных, но условия Неймана используют производную по направлению внешней нормали. Поэтому учитывайте знак производной по направлению при создании граничных условий в коде своей программы.
- Возможно, стоит начать с обучающей выборки в 5000-20000 точек.
- Если есть проблемы со сходимостью, то ресэмплирование точек обучающей выборки может помочь (например, каждые 100-1000 итераций): [https://deepxde.readthedocs.io/en/latest/demos/pinn\\_forward/heat.resample.html](https://deepxde.readthedocs.io/en/latest/demos/pinn_forward/heat.resample.html).
- Можно взять раз в 10 меньше точек для обучающей выборки, для условий PointSetBC задать `batch_size` сравнимого размера и задействовать ресемплирование точек, например, каждые 100 итераций для ускорения сходимости.
- Ищите значения параметров, к которым более-менее сходятся несколько запусков с достаточно разных начальных приближений. Например,  $\pm 0.5$ . Финальное значение  $\lambda$  можно вычислить усреднением значений, полученных с помощью разных начальных приближений.

- $\lambda \in [-30, 30]$ .
- Если будут проблемы, изучайте примеры на сайте и код на GitHub (следите, чтобы версия была как та, которую вы используете):
  - [https://deepxde.readthedocs.io/en/latest/demos/pinn\\_forward.html](https://deepxde.readthedocs.io/en/latest/demos/pinn_forward.html)
  - [https://deepxde.readthedocs.io/en/latest/demos/pinn\\_inverse.html](https://deepxde.readthedocs.io/en/latest/demos/pinn_inverse.html)
  - <https://github.com/lululxvi/deepxde/tree/v1.13.2>
- Если документация, код в репозитории и issues на GitHub и ваши товарищи не смогли вам помочь, то пишите мне, будем разбираться вместе.