ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ имени М.В.ЛОМОНОСОВА»

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

Решение уравнений в гидродинамике с помощью физически информированных нейронных сетей.

Выполнил: Родыгин Вадим Игоревич
студент 404 группы

Преподаватель: Приклонский Владимир Иванович

Москва
2024

Содержание

Введение

Уже долгое время в физике сплошных сред изучаются свойства поверхности жидкости: образование поверхностного слоя, теплообмен на поверхности, движение поверхности и др. На протяжении многих лет основной проблемой оставались измерения гидродинамических параметров в тонком слое на границе раздела сред — использование зондов как в работе [1] дает неточные результаты, так как вблизи зонда рельеф поверхности и течение меняются, и мы получаем информацию только в окрестности зонда. Тем не менее, в упомянутой выше работе с низкой точностью экспериментально были получены профили температур $T(z)$ в зависимости от глубины погружения зонда.

Первый неинвазивный метод, позволяющий получить информацию о поверхностном слое жидкости был продемонстрирован в работе [2]. Результаты этого эксперимента показали, что поверхность может быть как неподвижной, так и движущейся. Следовательно, в зависимости от изучаемой жидкости при численном моделировании требуются различные граничные условия для получения правильных результатов.

Повысить точность определения температуры в граничном слое можно при помощи численного моделирования. При совмещении результатов расчетов и экспериментальных данных можно получить большое количество информации о поверхностном слое. Однако при этом используются ресурсоемкие компьютерные методы (такие как схема конечных элементов) и большое количество приближений.

Актуальной для гидродинамики темой является использование нейросетей для решения гидродинамических систем уравнений и моделирования течений. Это позволяет экономить большое количество ресурсов, так как для решения похожих задач можно использовать одну и ту же заранее обученную нейросеть [3]. При этом обучать нейросеть можно на результатах классического численного моделирования, например на результатах моделирования по схеме конечных элементов. В частности, сейчас все чаще используются физически информированные нейросети, так как результаты вычис-

лений, полученные с их помощью, более предсказуемы. Такие нейросети отличаются от классических наличием в функции потерь, помимо квадратичной ошибки, физических уравнений. Такой подход приводит к более предсказуемым результатам работы нейросети, а также упрощает проверку и объяснение полученных зависимостей.

С математической точки зрения, нейросети решают задачу о нелинейной минимизации ошибки [4]. При этом для обучения классических нейросетей в рамках некоторой модели необходимо большое количество экспериментальных данных. Использование уравнений в функции потерь у физически информированных нейронных сетей позволяет сильно понизить количество необходимых экспериментальных точек, а также получить более физичные результаты.

В данной работе решается одно из важнейших уравнений в гидродинамических системах — уравнение переноса тепла. Для его решения будет использована физически информированная нейронная сеть. Обучение нейросети будет проводиться на точном решении, и затем будет проведено сравнение точного решения и решения, полученного нейросетью. Добавив в систему дополнительные параметры, такие как плотность и импульс, и уравнения на них можно полностью описать движение жидкости при различных условиях.

1 Аналитическое решение задачи

Перейдем к рассмотренью уравнения переноса тепла. Оно связано с уравнением переноса энергии, которое в свою очередь входит в систему, описывающую движение жидкости. Стандартное уравнение теплопроводности выглядит следующим образом:

$$-a^2 \Delta u + \frac{\partial u}{\partial t} = f, \tag{1}$$

где u — температура, t — время, f — неоднородность. Данная задача имеет точное аналитическое решение, которое, однако, представлено в виде рядов Фурье.

Рассмотрим простейшее пространство для уравнения — одномерное. Получим одномерное уравнение теплопроводности. После раскрытия оператора Лапласа оно будет иметь следующий вид:

$$-a^2 \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial t} = f(x). \tag{2}$$

Для наглядности работы метода решения задачи с использованием физически информированных нейронных сетей будем рассматривать установившийся режим. В таком режиме нет зависимости от времени. И таким образом мы сводим наше уравнение переноса тепла к дифференциальному уравнению второго порядка, которое имеет точное решение в квадратурах:

$$\frac{\partial^2 u}{\partial x^2} = f(x). \tag{3}$$

В литературе стационарное уравнение теплопроводности называют уравнением Пуассона [5], а однородное уравнение Пуассона (где $f = 0$) называют уравнением Лапласа. Эти уравнения хорошо изучены в электростатике, однако очень часто встречаются и в других областях.

Ограничим область рассмотрения по x:

$$x \in [0, L], \tag{4}$$

а также, как известно, для решения такого дифференциального уравнения необходимо поставить 2 граничных условия [6]. Например, пусть это будут условия Дирихле.

$$u(0) = u_1, \tag{5}$$

$$u(L) = u_L. \tag{6}$$

Система уравнений 3, 4, 5, 6 имеет точное аналитическое решение. Для примера возьмем неоднородность

$$f(x) = const = 4, \tag{7}$$

и получим аналитическое решение уравнения в этом случае:

$$u(x) = u_1 - u_1 x + \frac{x(-2 + a^2 u_L + 2x)}{a^2} \tag{8}$$

В дальнейшем необходимо будет подставить конкретные числа вместо a, u_1 и u_L, чтобы получить функцию, которую можно сравнить с результатами численного моделирования или с результатами, полученными при расчете с нейросетью.

2 Обучение физически информированной нейронной сети

В этом разделе мы переходим к численным расчетам. В аналитической задаче, построенной ранее, необходимо доопределить несколько констант. Возьмем значения

$$u_1 = 0, \tag{9}$$

$$u_L = 1, \tag{10}$$

$$a = 0.5; \tag{11}$$

и построим аналитическое решение в этом случае (Рис. 1). На графике расчетная сетка по оси x состоит из 500 точек, из них мы возьмем 10, откладывая их от начала отсчета с некоторым промежутком. Это точки, которые будут использованы для обучения нейронной сети.

Далее при вычислении ошибки укажем среднеквадратичное отклонение, но добавим к нему еще и ошибку, связанную с физическими уравнениями в модели. Формула ошибки будет выглядеть следующим образом:

$$MSE_{loss} = MSE_{regression} + MSE_{phys}; \tag{12}$$

ниже приведен программный код на Python для обучения нейросети, решающей поставленную выше задачу.

Для обучения нейросети используется фреймворк PyTorch, который позволяет быстро решать связанные с ними задачи. Подключаем необходимые библиотеки:

```
    from PIL import Image
    import numpy as np
    import torch
    import torch.nn as nn
    import matplotlib.pyplot as plt
```

Далее приведен код аналитического решения и созданный класс нейросети.

```
def save_gif_PIL(outfile, files, fps=5, loop=0):
  "Helper function for saving GIFs"
```

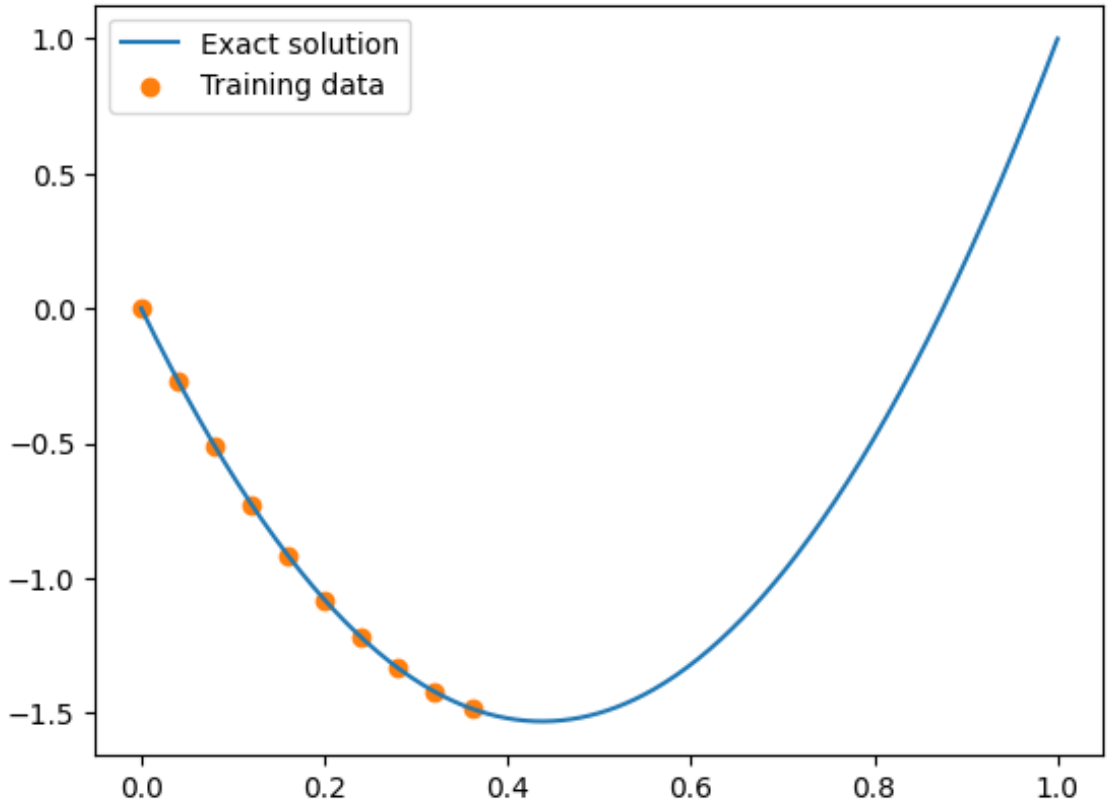


Рис. 1: Аналитическое решение

```
  imgs = [Image.open(file) for file in files]
  imgs[0].save(fp=outfile, format='GIF', append_images=imgs[1:],
   save_all=True, duration=int(1000/fps), loop=loop)

def heat_transfer(a, y1, yL, x):
  return y1 - y1*x + (x*(-2 + a**2*yL + 2*x))/a**2

class FCN(nn.Module):
  "Defines a connected network"

  def __init__(self, N_INPUT, N_OUTPUT, N_HIDDEN, N_LAYERS):
    super().__init__()
    activation = nn.Tanh
    self.fcs = nn.Sequential(*[
        nn.Linear(N_INPUT, N_HIDDEN),
        activation()])
    self.fch = nn.Sequential(*[
        nn.Sequential(*[
          nn.Linear(N_HIDDEN, N_HIDDEN),
          activation()]) for _ in range(N_LAYERS-1)])
    self.fce = nn.Linear(N_HIDDEN, N_OUTPUT)

  def forward(self, x):
    x = self.fcs(x)
```

```
    x = self.fch(x)
    x = self.fce(x)
    return x
```

Теперь задаем выбираем несколько тренировочных точек из аналитического решения и строим график, показанный на рисунке 1.

```
y1=0
yL=1
a=0.5

# get the analytical solution over the full domain
x = torch.linspace(0,1,500).view(-1,1)
y = heat_transfer(a,y1,yL,x).view(-1,1)
print(x.shape, y.shape)

# slice out a small number of points from the LHS of the domain
x_data = x[0:200:20]
y_data = y[0:200:20]
print(x_data.shape, y_data.shape)

plt.figure()
plt.plot(x, y, label="Exact solution")
plt.scatter(x_data, y_data, color="tab:orange", label="Training
    data")
plt.legend()
plt.show()
```

И последний этап — непосредственно обучение физически информированной нейросети с учетом формулы ошибки 12.

```
x_physics = torch.linspace(0,1,30).view(-1,1).requires_grad_(True)#
    sample locations over the problem domain

torch.manual_seed(123)
model = FCN(1,1,32,3)
optimizer = torch.optim.Adam(model.parameters(),lr=1e-4)
files = []

for i in range(20000):
  optimizer.zero_grad()

  # compute the "data loss"
  yh = model(x_data)
  loss1 = torch.mean((yh-y_data)**2) #mean squared error

  # compute the "physics loss"
  yhp = model(x_physics)
  dx = torch.autograd.grad(yhp, x_physics, torch.ones_like(yhp),
   create_graph=True)[0] # computes dy/dx
```

```
  dx2 = torch.autograd.grad(dx, x_physics, torch.ones_like(dx),
    create_graph=True)[0] # computes d^2y/dx^2
  physics = a**2*dx2 - 4 # computes the residual of the 1D harmonic
    oscillator differential equasion
  loss2 = (1e-4)*torch.mean(physics**2)

  # backpropagate joint loss
  loss = loss1 + loss2 #add two loss terms together
  loss.backward()
  optimizer.step()

  # plot the result as training progresses
  if (i+1) % 150 == 0:

    yh = model(x).detach()
    xp = x_physics.detach()

    plot_result(x,y,x_data,y_data,yh,xp)

    """file = "plots/pinn_%.8i.png"%(i+1)
    plt.savefig(file, bbox_inches='tight', pad_inches=0.1, dpi=100,
    facecolor="white")
    files.append(file)"""

    if (i+1) % 6000 == 0: plt.show()
    else: plt.close("all")

save_gif_PIL("pinn.gif", files, fps=20, loop=0)
```

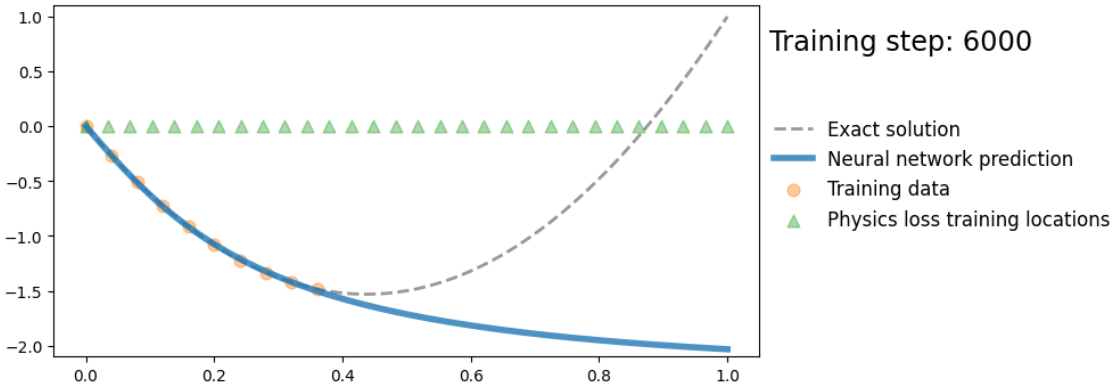
3 Результаты

В результате работы программы были получены графики, показывающие отличие предсказанного нейросетью решения от аналитического.

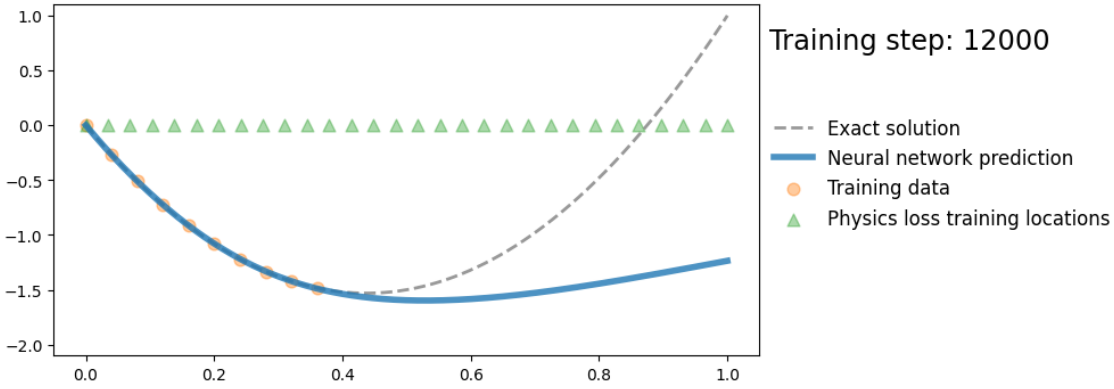
Как видно из графиков (Рис. 2), при достаточном количестве шагов обучения нейросеть может дать решение поставленной задачи с большой точностью. Очевидно, точность решения растет с каждым шагом. При этом полученное решение гладкое, в нем не возникает непредвиденных перегибов или осцилляций. Оно легко поддается анализу и удовлетворяет физическим уравнениям.

Физически информированная нейросеть требует меньшее число обучающих шагов, как показано в [4] по сравнению с классической нейронной сетью. Такое преимущество позволяет дополнительно уменьшить затраты на обучение.

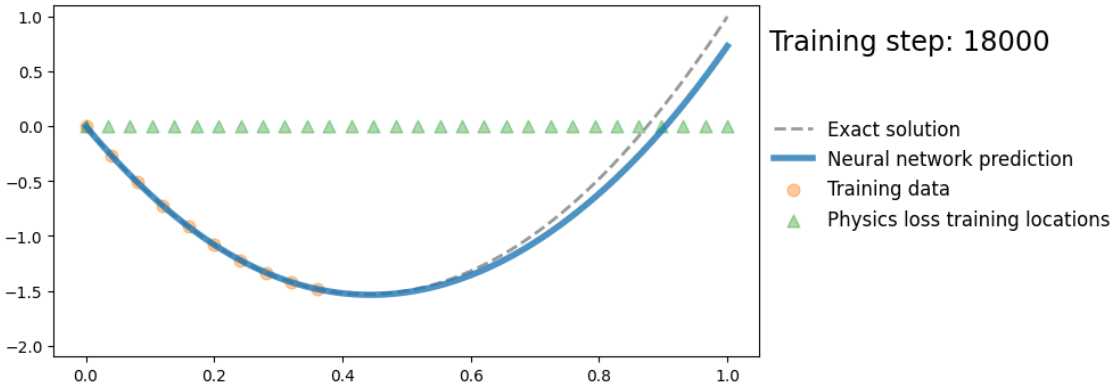
Если бы обучающие точки взять не из аналитического решения, а из эксперимента, то после подстановки полученного нейросетью решения в исходные уравнения можно получить неизвестные коэффициенты. Таким образом, физически информированные нейросети позволяют подобрать неизвестные коэффициенты в системе.



(a) Предсказания нейросети на шаге номер 6000



(b) Предсказания нейросети на шаге номер 12000



(c) Предсказания нейросети на шаге номер 18000

Рис. 2: Предсказания нейросети

4 Выводы

Использование нейросетей значительно сокращает расчетное время, требуемое компьютером на решение численной задачи. Тем не менее, необходимо затратить время на ее обучение, на что очень часто при решении численных задач уходит время, не меньшее, чем тратится на прямой расчет. Однако, у такого подхода есть свои преимущества. После обучения нейросети она сможет быстро решать однотипные задачи. Поэтому, ее использование может быть оправдано в таких случаях.

В естественно-научных областях сейчас активно развиваются физически информированные нейронные сети. Они зарекомендовали себя как нейронные сети, позволяющие работать в рамках конкретной физической модели и получать решения, которые соответствуют всем уравнениям. Их решения проще поддаются объяснению и чаще всего имеют прямой физический смысл.

В области гидродинамики очень редко удается получить аналитическое решение, описывающее движение жидкости. Для решения системы уравнений жидкости используется большое количество приближений и ресурсоемкие компьютерные методы. Здесь на помощь могут прийти физически информированные нейронные сети, так как они позволяют решать однотипные задачи с гораздо меньшими затратами ресурсов компьютера [3].

При достаточно долгом обучении физически информированной нейронной сети, имея всего несколько точек, где известно решение и уравнения, описывающие систему, можно получить достаточно точное решение на всей числовой прямой. При этом решения физически информированной нейронной сети сходятся к реальным быстрее, чем решения, полученные с помощью классических нейронных сетей.

Список литературы

[1] K. B. Katsaros, W. T. Liu, J. A. Businger, and J. E. Tillman, "Heat thermal structure in the interfacial boundary layer measured in an open tank of water in turbulent free convection," *Journal of Fluid Mechanics*, vol. 83, no. 2, pp. 311–335, 1977.

[2] W. Spangenberg and W. Rowland, "Convective circulation in water induced by evaporative cooling," *The Physics of Fluids*, vol. 4, no. 6, pp. 743–750, 1961.

[3] F. Darlik and B. Peters, "Reconstruct the biomass particles fields in the particle-fluid problem using continuum methods by applying the physics-informed neural network," *Results in Engineering*, vol. 17, p. 100917, 2023.

[4] S. Pratik and R. Iriondo, "Neural networks from scratch with python code and math in detail — i," *Towards AI*, 2020.

[5] T. Fukuchi, "A whole high-accuracy numerical calculation system for the 1d poisson equation by the interpolation finite difference method," *AIP Advances*, vol. 12, no. 10, 2022.

[6] Рындин, Куликова, and Лысенко, "Основы численных методов: теория и практика," *Электронное учебное пособие. Таганрог*, 2015.