

有限差分法和PINN法求解微分方程的探讨

王 玮, 唐 虹, 张亭亭, 梁育境, 侯玉霞, 李萌慧, 张运章*

河南科技大学数学与统计学院, 河南 洛阳

收稿日期: 2023年6月21日; 录用日期: 2023年7月16日; 发布日期: 2023年7月25日

摘 要

在工程实际中的许多问题最终都可以转化为微分方程。由于一些微分方程复杂性, 这些方程求解通常具有一定的难度。随着计算机的迅速发展, 使得这些方程可以数值求解。如何设计高效的微分方程数值解法尤其重要。微分方程数值解法通常包括有限差分、有限元、有限体积等。近年来基于深度学习的微分方程求解方法十分火热。本文对内嵌物理信息神经网络(PINN)方法进行探讨。我们用传统的有限差分法和PINN法对常微分两点边值问题和偏微分方程中的一类热传导方程进行数值求解, 对比分析两种数值解法的优缺点。从数值实验结果中可以看出用PINN相对于传统有限差分法求解微分方程具有更好的精度和效率。

关键词

微分方程, 神经网络, 物理信息神经网络(PINN), 有限差分法

Discussion on Solving Differential Equations by Finite Difference Method and PINN Method

Wei Wang, Hong Tang, Tingting Zhang, Yujing Liang, Yuxia Hou, Menghui Li, Yunzhang Zhang*

School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang Henan

Received: Jun. 21st, 2023; accepted: Jul. 16th, 2023; published: Jul. 25th, 2023

Abstract

Many problems in engineering practice can ultimately be transformed into differential equations.

*通讯作者。

文章引用: 王玮, 唐虹, 张亭亭, 梁育境, 侯玉霞, 李萌慧, 张运章. 有限差分法和 PINN 法求解微分方程的探讨[J]. 应用数学进展, 2023, 12(7): 3298-3310. DOI: 10.12677/aam.2023.127329

Due to the complexity of some differential equations, solving these equations usually has some difficulty. With the rapid development of computers, these equations can be numerically solved. How to design efficient numerical solutions for differential equations is particularly important. The numerical solution of differential equations usually includes finite difference, finite element, finite volume, etc. In recent years, differential equation solving methods based on deep learning have become very popular. This article explores the embedded Physics-Informed Neural Networks (PINN) method. We use the traditional finite difference method and the PINN method to numerically solve the ordinary differential two-point boundary value problem and a kind of heat conduction equation in the partial differential equation, and compare the advantages and disadvantages of the two numerical methods. From the numerical experiment results, it can be seen that PINN has better accuracy and efficiency than the traditional finite difference method in solving differential equations.

Keywords

Differential Equations, Deep Neural Network, Physics-Informed Neural Networks (PINN), Finite Difference Method

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言和方法介绍

1.1. 微分方程

微分方程(differential equation)是含有未知函数及其导数(或偏导数)的等式。微分方程包含常微分方程和偏微分方程两类。微分方程广泛应用于物理、工程、生命科学、经济学等领域。例如,在过去战争方式单一时期,微分方程数学模型可以很好地预测战争的局势;而在经济学理论中,微分方程可以分析商品市场价格和需求量的关系,预测和分析一些经济问题;在医学中,微分方程描述事物的动态过程,可以揭示疾病的一些规律,如传染病模型、图像处理[1]、癌分析等都用到微分方程。由于方程以及边界条件的复杂性,很难得到微分方程的精确解(解析解),因此微分方程的数值求解很有必要。

常微分方程数值求解分为初值问题、边值问题和初值边值混合问题。本文以常微分两点边值问题为例进行讨论。它的数值解法主要有:有限差分法、有限元法、龙格库塔法[2]、辛普森法则、伽辽金方法等,它们有不同的特点,我们根据问题选择相应的方法,如果求解的问题较简单,可以用有限差分法或者辛普森法则;如果问题较复杂,可以用有限元法或者龙格-库塔法。偏微分方程传统数值解法主要有:有限差分法、有限元法、有限体积法、边界元法、谱方法、径向基函数方法等。我们要根据问题对这些方法进行选择。如果求解问题简单,就用有限差分法;如果需要高精度和快速收敛速度,就用谱方法或径向基函数方法;如果需要处理复杂的几何形状,就用有限元法或边界元法。本文采用有限差分法。

近年来,通过深度学习求解微分方程愈加火热化,出现的一些比较流行的方法有:物理信息神经网络(PINN) [3]、神经常微分方程(NODE)、残差神经网络(ResNet)、生成对抗网络(GAN)、变分自编码器(VAE)、卷积神经网络(CNN)等,它们都有各自的优缺点、应用领域和局限。方法的选择也取决于具体问题和可用资源。

本文以内嵌物理信息神经网络 PINN 方法求解微分方程进行探讨。

1.2. 有限差分法

有限差分法是一种求解微分方程的数值解法。它的原理是利用网格节点逼近导数，并建立有限个未知数的代数方程组来求解各个网格上节点的值。在有限差分法中，我们首先将求解域分成一个个网格单元，每个网格单元上的函数值用一个节点来代表。然后，我们用中心差分公式或向前、向后差分公式来逼近微分方程中的导数，从而得到代表微分方程的离散方程组。通过求解这个离散方程组，我们可以得到每个节点上的函数值，从而得到整个求解域上的函数解。几种常见的差分格式有：

一阶差分格式：

$$\frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x)}{h}, \text{ 向前差分} \tag{1}$$

$$\frac{\Delta f(x)}{\Delta x} = \frac{f(x) - f(x-h)}{h}, \text{ 向后差分} \tag{2}$$

$$\frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x-h)}{2h}, \text{ 中心差分} \tag{3}$$

二阶差分格式：

$$\frac{\Delta^2 f(x)}{\Delta x^2} = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{4}$$

具体求解步骤可见数值实验部分。那我们就可以知道它的核心是使用差分近似微分，这个过程要舍掉截断误差，预示就产生误差，且无法避免，这个误差量级与选择的差分格式有关。有限差分法的优缺点如下图 1：

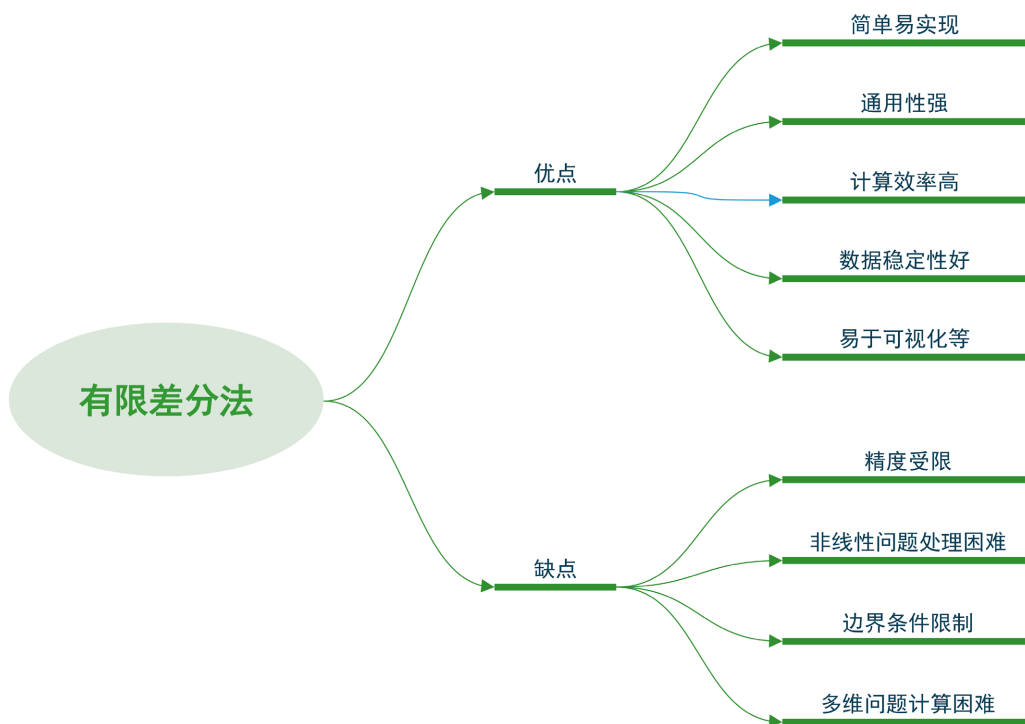


Figure 1. Plot of advantages and disadvantages of finite difference method
图 1. 有限差分法优缺点图

1.3. 深度学习神经网络

深度学习神经网络是一种用于机器学习和人工智能的算法。它由多个层次的神经元组成，可以对大量数据进行训练，以识别图像、语音、文本等信息。

深度学习神经网络的基本结构如图 2 所示，包括输入层、隐藏层、输出层，是一个前向传播的多层神经网络。在这个网络中，输入数据通过多个隐藏层逐步传递，并被转化为输出结果。每个隐藏层包含多个神经元，每个神经元与上一层的所有神经元相连。每个连接都有一个权重值，用于控制信息流的强度。

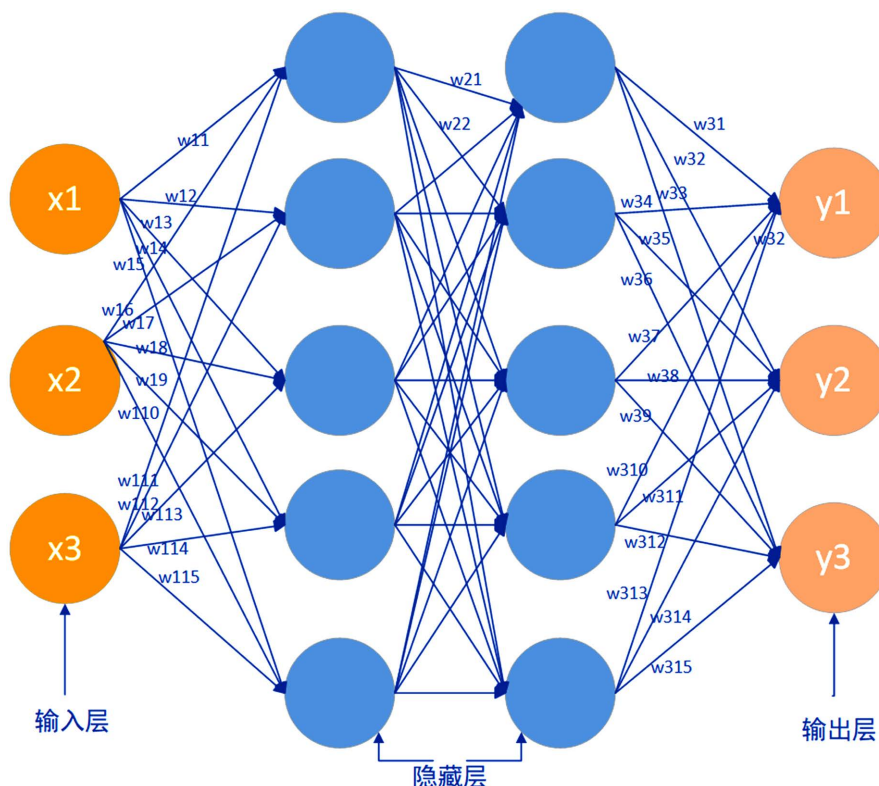


Figure 2. Neural network architecture diagram
图 2. 神经网络结构图

训练深度学习神经网络通常使用反向传播[4]算法。该算法通过比较预测结果与真实结果之间的误差来调整每个连接的权重。通过多次迭代训练，网络逐渐学会提取特征并生成准确的预测结果。

深度学习神经网络已被广泛应用于计算机视觉，如服装图像识别[5]，自然语言处理，物理信息预测，如金融预测[6]，风电功率预测[7]等。

物理信息神经网络(PINN)

PINN (Physics-Informed Neural Networks)是一类用于解决有监督学习任务存在的神经网络，可以在解决物理问题时提高精度和效率。它结合了神经网络和物理方程，并使用已知的物理规律作为约束条件来训练模型。基本原理是将微分方程中的物理约束嵌入到神经网络的训练过程中，以提高模型的精度和泛化能力。PINN 使用神经网络来近似未知的物理场，同时也能够满足已知的物理方程，因此可以减少计算成本并提高求解精度。具体地，PINN 将物理方程表示为损失函数，并通过反向传播算法来调整神经网络

的参数,以最小化损失函数,在某些情况下它比传统方法更准确和高效。PINN 的关键在于将物理约束嵌入到神经网络的训练过程中。这样做可以使模型更好地符合实际问题的物理规律,并提高模型的精度和泛化能力。同时, PINN 也可以自动处理一些传统数值方法中需要手动调整的参数,例如网格大小和时间步长等。因此, PINN 可以在求解微分方程方面具有更好的效率和精度。

但它也存在一些缺点,包括:训练时间长、精度受限、需要大量数据、可解释性差、超参数敏感性等。

2. 常微分两点边值问题数值求解

我们分别用有限差分法和 PINN 对常微分两点边值问题数值求解,对比了两个方法的精度和效率。

2.1. 有限差分法求解

两点边值问题的一般形式

$$\begin{cases} \frac{d^2 y}{dx^2} + p(x) \frac{dy}{dx} + q(x)y = f(x), & 0 < x < l \\ y(0) = \alpha, & y(l) = \beta \end{cases} \quad (5)$$

求解步骤(四步法):

① 网格剖分 将 $[0, l]$ 剖成 $N+1$ 等份,步长为 $h=l/(N+1)$,节点 $x_i = ih, i=0, 1, \dots, N+1$,

② 将节点 (x_i, t_j) 代入方程得

$$y''(x_i) + p(x_i)y'(x_i) + q(x_i)y(x_i) = f(x_i) \quad (6)$$

③ 用差分代替微分,用二阶中心差分和一阶中心差分代替微分即

$$y''(x_i) = \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} + o(h^2) \quad (7)$$

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1}))}{2h} + o(h^2) \quad (8)$$

带入方程我们得

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} + p(x_i) \frac{y(x_{i+1}) - y(x_{i-1}))}{2h} + q(x_i)y(x_i) + o(h^2) = f(x_i) \quad (9)$$

$$y(x_0) = \alpha, \quad y(x_{N+1}) = \beta$$

④ 舍掉截断误差我们得到:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i) \frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y(x_i) = f(x_i) \quad (10)$$

整理以后得到:

$$\begin{aligned} c_i y_{i+1} + a_i y_i + b_i y_{i-1} &= h^2 f(x_i), \quad i=1, 2, \dots, N \\ y_0 &= \alpha, \quad y_{N+1} = \beta \end{aligned} \quad (11)$$

其中

$$a_i = -2 + h^2 q(x_i), \quad b_i = 1 - \frac{h}{2} p(x_i), \quad c_i = 1 + \frac{h}{2} p(x_i)$$

因此我们得矩阵方程

通过图 3, 我们发现: 随着 $N=9$, $N=19$, $N=49$, 剖分份数的增加, 求解结果与精确解越来越接近, 在 $x=1.5$ 附近, $N=9$ 求解结果与精确解误差较大; 从整体来看在区间两端往区间中间, 数值解与精确解之间的误差越来越大, 我们再来看 $x=1.5$ 附近求解结果的局部放大, 如图 4 所示:

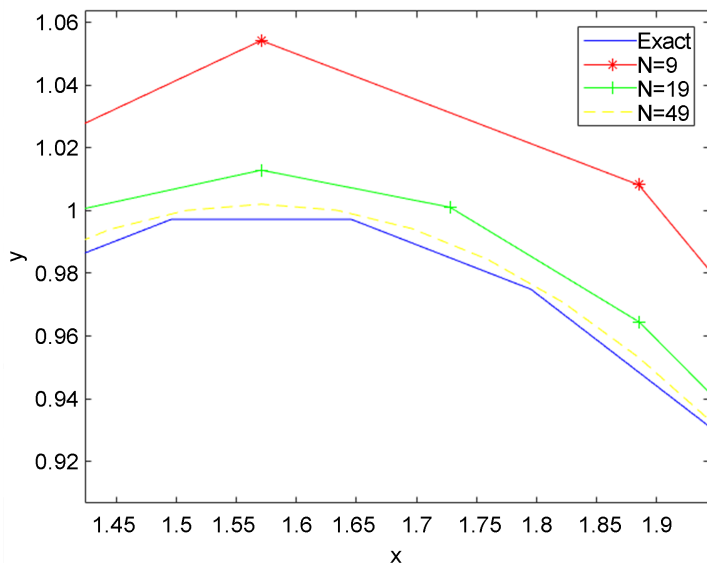


Figure 4. Programming the two-point edge value problem solves the result detail plot

图 4. 两点边值问题编程求解结果细节图

从局部放大图 4 我们可以更加清晰地观察到随着剖分份数 N 的增加, 有限差分求得的数值解与精确解之间的误差越来越小。 $N=49$ 的数值逼近效果最好, $N=19$ 的数值逼近效果次之, $N=9$ 的数值逼近效果最差。特别地, 误差最大达到 10^{-2} 量级。整体而言数值方法的收敛性良好。同时, 使用追赶法求解大规模线性方程组的速度相对较快, 但整体误差在 $o(h^2)$ 量级。

2.2. PINN 求解

PINN 通过以下几个步骤来求解微分方程:

1) 将微分方程转换为损失函数: 将微分方程中的偏微分算子表示为神经网络的输出, 并将微分方程中的初始条件和边界条件表示为神经网络的输入。然后, 将微分方程和边界条件转换为一个损失函数, 用于指导神经网络的训练。

2) 训练神经网络: 使用反向传播算法和优化器来最小化损失函数, 以更新神经网络的权重和偏置参数。在此过程中, 神经网络将自动学习微分方程的解, 并逐渐提高模型的精度和泛化能力。

3) 检验结果: 使用训练好的神经网络来预测微分方程的解, 并与真实解进行比较。如果误差较小, 则可以认为神经网络求解了微分方程。

$$\begin{cases} y'' - y' + y = -\cos x, & 0 < x < \pi \\ y(0) = 0, & y(\pi) = 0 \end{cases} \quad (14)$$

我们用 PINN 再对上面这个常微分两点边值问题进行求解。我们设定一个输入层、3 个包含 20 个神经元的隐藏层和一个输出层, 使用双曲正切激活函数和 Glorot 正态分布初始化方法初始化权重, 设定我们的问题域采样点数量为 2000, 边界采样点数量为 10, 测试数据集采样点数量为 1600, 对我们的模型

进行 10,000 次训练周期，使用默认的 L-BFGS-B 优化器进行微调，绘制出损失历史和训练状态，然后预测出我们的方程区间 $[0, \pi]$ 均匀间隔 101 个点的预测解和残差，训练次数对应的训练损失和测试损失结果见表 1。

Table 1. Training loss and test loss corresponding to the number of trainings

表 1. 训练次数对应的训练损失、测试损失

Step	Train loss	Test loss
0	[2.78e-01, 3.95e-02]	[2.76e-01, 3.95e-02]
1000	[9.28e-05, 4.41e-06]	[8.98e-05, 4.41e-06]
2000	[1.36e-05, 7.99e-07]	[1.32e-05, 7.99e-07]
3000	[7.51e-06, 5.29e-07]	[7.28e-06, 5.29e-07]
4000	[5.12e-06, 1.98e-07]	[4.96e-06, 1.98e-07]
5000	[3.61e-06, 6.31e-08]	[3.50e-06, 6.31e-08]
6000	[2.46e-06, 2.78e-08]	[2.38e-06, 2.27e-08]
7000	[1.70e-06, 6.95e-09]	[1.64e-06, 6.95e-09]
8000	[1.26e-06, 3.64e-10]	[1.21e-06, 3.64e-10]
9000	[1.39e-05, 4.03e-05]	[1.35e-05, 4.03e-05]
10,000	[8.96e-07, 1.43e-08]	[8.49e-07, 1.43e-08]
10,122	[6.43e-08, 1.27e-10]	[6.10e-08, 1.27e-10]

从这个表我们可以看出：随着训练周期的增加，我们的训练损失、测试损失越小。而其中最好的模型是设置训练次数为 10,122 时，它的训练损失为 $6.44e-08$ ，测试损失为 $6.12e-08$ ，训练用时为 0.904711 s。而在训练次数为 10,000 时，我们的训练损失和测试损失已经可以达到 10^{-7} 量级。具体计算结果见图 5。

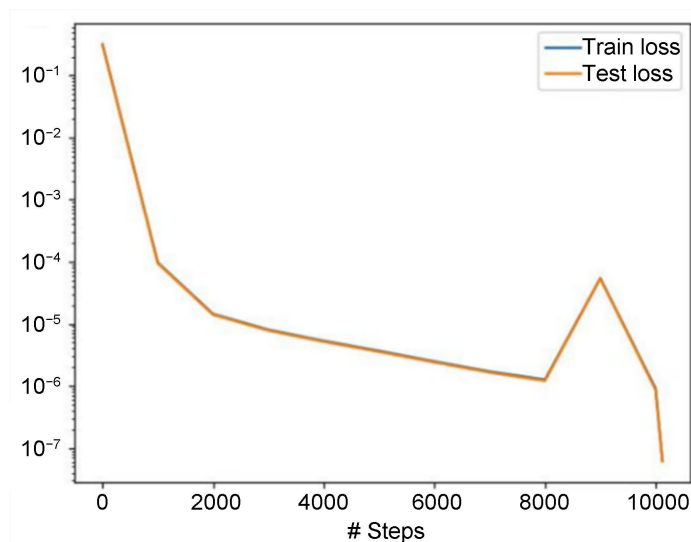


Figure 5. Training and testing loss plots for PINN solving equations

图 5. PINN 求解方程的训练损失、测试损失图

从图 5 我们也可以直观地看出迭代次数小于 8000 时，随着迭代次数的增加，训练损失和测试损失越来越小。而训练次数在 8000 至 10,000 之间，训练损失和测试损失进行了一个先增后减的趋势，到迭代

次数为 10,000 时，我们的训练损失和测试损失达到了最小，误差大致在 10^{-7} 量级，结果与我们的表格结果一致。

设定迭代次数为 10,000，计算机求解时间是 0.214267 s，再来看我们求得的预测解、方程的精确解以及它们之间的误差，见图 6。

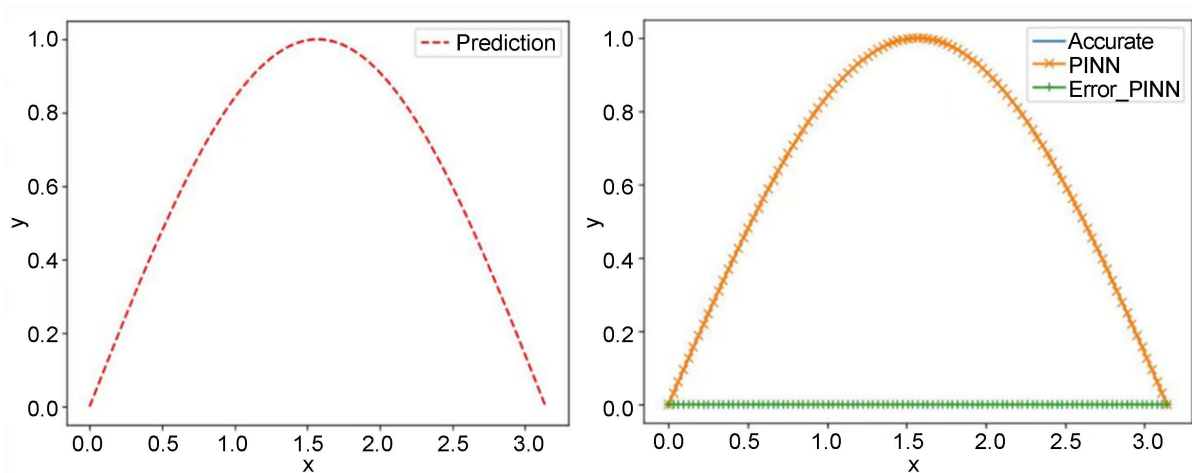


Figure 6. The graph of the predicted solution obtained by PINN, the exact solution of the equation, and the error between them

图 6. PINN 求得的预测解、方程的精确解及它们之间的误差图

图 6 的左图是我们用 PINN 求得的预测解图像，图 6 的右图为精确解、PINN 求得解以及它们之间的误差图。我们可以看出右图中 PINN 数值解已经把精确解覆盖了。从右图我们也可以看出通过 PINN 计算的结果和精确解很接近，误差几乎为 0。由表 1 我们知道 PINN 的误差量级为 10^{-7} ，而有限差分法误差量级为 10^{-2} ，故 PINN 方法比传统有限差分计算精确，但是 PINN 方法的求解速度上比传统有限差分慢。

综上所述：求解该问题，PINN 方法在精度方面要优于传统的有限差分法，但在计算效率(时间)方面不如传统的有限差分法。

3. 热传导方程数值求解

本节，我们分别用有限差分法和 PINN 对热传导方程数值求解，对比了两个方法的精度和计算效率。

3.1. 有限差分法求解

热传导方程[8]

$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2} + f(x), 0 < t \leq T \tag{15}$$

初始条件:

$$u(x, 0) = g(x), 0 < x < 1$$

边界条件:

$$u(0, t) = u(1, t) = 0, 0 \leq t \leq T$$

求解步骤:

① 网格剖分 将 $[0,1]$ 剖成 $N + 1$ 等份，步长为 $h = 1/(N + 1)$ ，节点 $x_i = ih, i = 0, 1, \dots, N + 1$ ，将 $[0, T]$ 剖

成 M 等份, 步长 $\tau = T/M$, 节点 $t_j = ik, i = 0, 1, \dots, M$ 。

② 将节点 (x_i, t_j) 代入方程得

$$u_t(x_i, t_j) = au_{xx}(x_i, t_j) + f(x_i, t_j) \quad (16)$$

③ 用时间向前空间中心差分代替微分

$$u_t(x_i, t_j) = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\tau} + o(\tau), \quad u_{xx}(x_i, t_j) = \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{h^2} + o(h^2) \quad (17)$$

代入得

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\tau} + o(\tau) = a \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{h^2} + o(h^2) + f(x_i, t_j) \quad (18)$$

④ 舍掉截断误差可得到如下的差分方程

$$\text{令 } \lambda = \frac{a\tau}{h^2}, \text{ 得 } u_{i,j+1} = \lambda u_{i+1,j} + (1-2\lambda)u_{i,j} + \lambda u_{i-1,j} + \tau f_{i,j} \quad (19)$$

$$i = 1, 2, \dots, N; j = 0, 1, \dots, M-1;$$

$$u_{i,0} = g_i, i = 1, 2, \dots, N+1, u_{0,t} = u_{N+1,t} = 0, t = 1, 2, \dots, M$$

因此

$$u_{j+1} = Au_j + \tau f_j, j = 0, 1, 2, \dots, M-1 \quad (20)$$

$$A = \begin{pmatrix} 1-2\lambda & \lambda & & & \\ \lambda & 1-2\lambda & \lambda & & 0 \\ & \lambda & 1-2\lambda & \lambda & \\ & & \ddots & \ddots & \ddots \\ 0 & & & & \lambda \\ & & & & \lambda & 1-2\lambda \end{pmatrix},$$

$$u_j = \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{N,j} \end{pmatrix}, f_j = \begin{pmatrix} f_{1,j} \\ f_{2,j} \\ \vdots \\ f_{N,j} \end{pmatrix}, g = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{pmatrix}$$

使用追赶法编程求解。它的误差为 $o(h^2) + o(\tau)$ 量级。

例 3.1 热传导方程初边值问题:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 2, & 0 < x < 1, 0 < t < T, \\ u(x, 0) = \sin(\pi x) + x(1-x), & 0 \leq x \leq 1, \\ u(0, t) = u(1, t) = 0, & t > 0. \end{cases} \quad (21)$$

对于这个问题我们先用有限差分法求解。我们采用时间向前, 空间中心的差分格式来求解该问题, 并与精确解 $u(x, t) = e^{-t^2} \sin(\pi x) + x(1-x)$ 比较。选取时间步长为 0.005, 空间步长为 0.1, 进行求解。

此时 $l = 1$, $N + 1 = 10$, 空间步长 $h = 0.1$, $T = 0.1$, $M = 20$, 时间步长 $\tau = 0.005$

$$a = 1, f(i, j) = 2, g(x) = \sin(\pi x) + x(1-x)$$

差分方程为:

$$u_{i,j+1} = \lambda u_{i+1,j} + (1 - 2\lambda)u_{i,j} + \lambda u_{i-1,j} + 2\tau, \lambda = \frac{\tau}{h^2} = 0.5 \tag{22}$$

$t = 0.1$ 时, 总的求解时间是 0.090691 s。我们编程求解结果为如图 7:

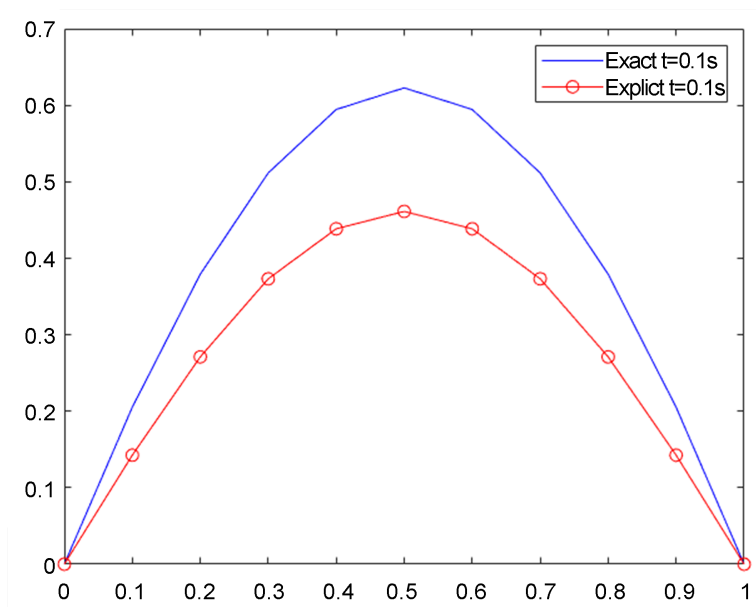


Figure 7. Plot of the analytical solution and exact solution obtained by finite difference method at $t = 0.1$ s

图 7. $t = 0.1$ s 时有限差分法求得的解析解和方程精确解的图

我们把利用有限差分法求得的数值解与该问题的精确解放到一起, 即图 7。从图 7 很容易可以发现, 数值解和精确解之间的误差在区间中间, 即在点 $x = 0.5$ 附近, 达到最大, 大致在 10^{-1} 量级。

3.2. PINN 求解

同样我们用 PINN 再对上面这个热传导问题(21)进行求解。我们设定 2 个输入层、3 个包含 20 个神经元的隐藏层和一个输出层, 使用双曲正切激活函数和 Glorot 正态分布初始化方法初始化权重, 设定我们的问题域采样点数量为 2500, 边界采样点数量为 1000, 测试数据集采样点数量为 1600, 对我们的模型进行 10,000 次训练周期, 设定优化器为 Adam 指定学习率为 1×10^{-3} , 设定迭代次数 1000 训练模型, 再设定优化器 L-BFGS-B 进一步提高精度, 绘制出损失历史和训练状态, 然后预测出我们的方程区间 $[0, \pi]^2$ 上均匀间隔 101 个点在 $t = 0.1$ s 的预测解和残差。训练次数对应的训练损失和测试损失结果见表 2。

Table 2. Training loss and test loss corresponding to the number of trainings

表 2. 训练次数对应的训练损失、测试损失

Step	Train loss	Test loss
0	[7.11e+00, 3.42e-01, 1.43e+00]	[7.11e+00, 3.42e-01, 1.43e+00]
1000	[6.59e-03, 1.26e-02, 3.96e-02]	[6.59e-03, 1.26e-02, 3.96e-02]
2000	[3.80e-03, 5.79e-03, 1.06e-02]	[3.80e-03, 5.79e-03, 1.06e-02]
3000	[1.63e-03, 1.88e-03, 1.02e-03]	[1.63e-03, 1.88e-03, 1.02e-03]

Continued

4000	[6.06e-04, 4.11e-04, 9.07e-05]	[6.06e-04, 4.11e-04, 9.07e-05]
5000	[2.98e-04, 1.38e-04, 2.15e-05]	[2.98e-04, 1.38e-04, 2.15e-05]
6000	[1.74e-04, 7.50e-05, 1.04e-05]	[1.74e-04, 7.50e-05, 1.04e-05]
7000	[1.15e-04, 4.47e-05, 5.98e-06]	[1.15e-04, 4.47e-05, 5.98e-06]
8000	[8.19e-05, 2.73e-05, 3.76e-06]	[8.19e-05, 2.73e-05, 3.79e-06]
9000	[6.02e-05, 1.66e-05, 2.40e-06]	[6.02e-05, 1.66e-05, 2.40e-06]
10,000	[4.96e-05, 1.04e-05, 1.51e-06]	[4.96e-05, 1.04e-05, 1.51e-06]
11,000	[8.73e-07, 1.40e-07, 4.55e-08]	[8.73e-07, 1.40e-07, 4.55e-08]
11,237	[5.97e-07, 3.01e-08, 4.39e-09]	[5.97e-07, 3.01e-08, 4.39e-09]

从表 2 我们可以看出：随着训练周期的增加，训练损失和测试损失越小。而其中最好的模型是设置训练次数为 11,237 时，训练损失为 $6.32e-07$ ，测试损失为 $6.32e-07$ ，训练用时为 0.904711 s。而在训练次数为 10,000 时，训练损失和测试损失可以达到 10^{-5} 量级。具体结果见图 8。

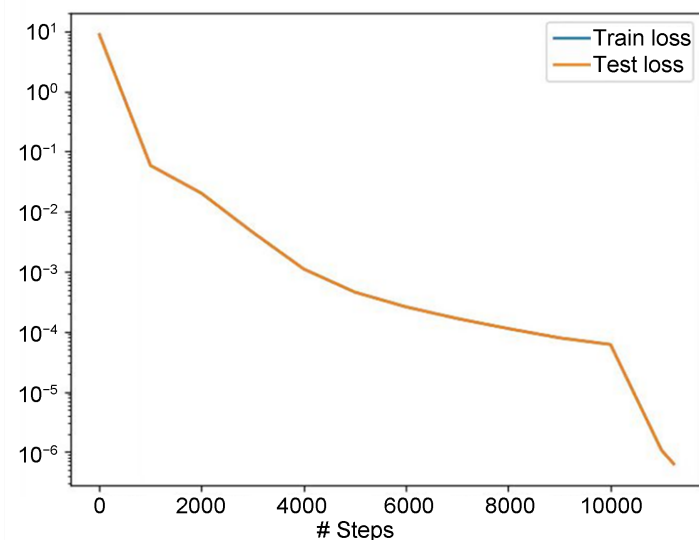


Figure 8. Training and testing loss plots for PINN solving equations
图 8. PINN 求解方程的训练损失、测试损失图

从图 8 很容易看出，随着迭代次数的增加，训练损失和测试损失越来越小。迭代次数在 10,000 次时，损失误差大致是 10^{-4} 量级，当迭代次数为 11,000 时，可以达到 10^{-6} 量级。

设定迭代次数为 10,000 时，计算机求解时间为 0.143290 s。再来看我们用 PINN 求得的预测解、方程的精确解以及它们之间的误差图(图 9)。

图 9 的左图是利用 PINN 方法求得的预测解图像，图 9 右图为精确解、PINN 求得解以及它们之间的误差图。我们很容易发现，通过 PINN 方法计算的数值解已经把精确解覆盖了。从右图我们同样可以看出：PINN 算得的结果和精确解很接近，误差几乎为 0，由表 2 我们知道，PINN 方法的计算误差量级为 10^{-5} ，而有限差分法误差量级为 10^{-1} 。因此 PINN 法比传统差分法求解精度高，但求解速度不如传统差分法。

综上所述，数值求解热传导方程时，PINN 方法在精度上要优于传统有限差分法求解，但在求解效率方面不如传统的有限差分法。

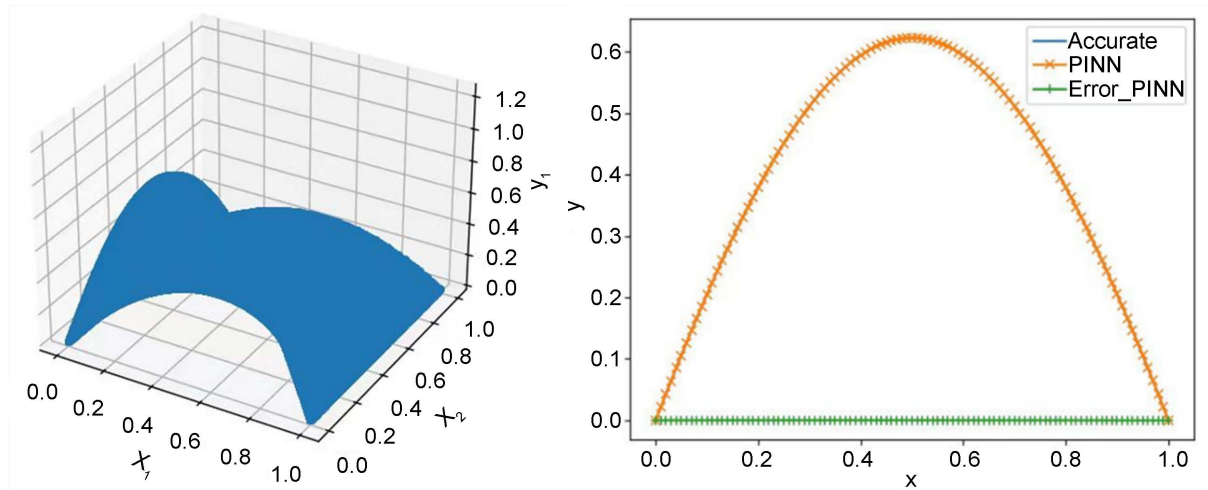


Figure 9. The graph of the predicted solution obtained by PINN, the exact solution of the equation, and the error between them

图 9. 通过 PINN 获得的预测解的图形、方程的精确解以及它们之间的误差

4. 结论与展望

本文基于有限差分法和 PINN (物理信息神经网络)方法,以常微分两点边值问题和热传导方程为例,探讨了这两种方法的优缺点,结果表明有限差分法求解速度相对 PINN 求解要快,但是精度不如 PINN 方法。基于此,针对不同的微分方程,我们可以根据不同需求选择对应的方法求解。对精度要求高的问题,可以使用 PINN 法,而对于精度要求不是很高,对求解速度比较高的时候,我们可以采用有限差分法进行求解。

基金项目

河南省高等学校重点科研项目(编号: 21B110003), 河南科技大学实验技术开发基金项目(编号: SY2223040), 河南科技大学教改项目(编号: 2021KB165), 河南科技大学大学生研究训练计划 SRTP (编号: 2022226, 2023226)。

参考文献

- [1] 孙志强. 微分方程在图像去噪处理中的应用研究[J]. 信息记录材料, 2022, 23(2): 145-147.
- [2] 胡庆婉. 常微分方程初值问题的数值求解及 MATLAB 实现[J]. 科技信息, 2012(7): 34-35.
- [3] Nascimento, R.G., Fricke, K. and Viana, F.A.C. (2020) A Tutorial on Solving Ordinary Differential Equations Using Python and Hybrid Physics-Informed Neural Network. *Engineering Applications of Artificial Intelligence*, **96**, Article ID: 103996. <https://doi.org/10.1016/j.engappai.2020.103996>
- [4] Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. *Neural Networks*, **61**, 85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [5] 曹敏. 基于深度学习的遥感图像识别技术应用研究——评《基于深度神经网络的遥感图像分割》[J]. 有色金属工程, 2021, 11(12): 132.
- [6] 张贵勇. 改进的卷积神经网络在金融预测中的应用研究[D]: [硕士学位论文]. 郑州: 郑州大学, 2016.
- [7] 刚毅凝, 金成明, 丁一, 等. 神经网络在风电功率预测中的应用研究[J]. 信息技术, 2020, 44(1): 150-152+158. <https://doi.org/10.13274/j.cnki.hdzj.2020.01.032>
- [8] 陈金雄, 张敏, 沈丹梅, 等. 有限差分法的一维热传导方程应用[J]. 武夷学院学报, 2022, 41(3): 53-57.