

基于两重网格的深度学习方法求解定常偏微分方程*

彭湃, 冯新龙[†]

(新疆大学 数学与系统科学学院, 新疆 乌鲁木齐 830017)

摘要: 随着机器学习在多个领域的研究取得进展, 物理信息神经网络为偏微分方程的求解提供了新思路, 但该方法难以获得高精度的数值解. 结合物理信息神经网络与两重网格求解偏微分方程的思想, 提出了基于两重网格的深度学习方法求解定常偏微分方程. 针对神经网络求解多目标问题, 采取了动态权重策略平衡损失函数中各项之间的数值差异, 有效缓解了梯度病态现象. 最后, 给出了若干数值实验, 验证了结合动态权重策略的深度学习方法在提高计算精度上的有效性.

关键词: 定常偏微分方程; 深度学习; 动态权重; 高精度; 两重网格

DOI: 10.13568/j.cnki.651094.651316.2021.09.22.0003

中图分类号: O242.21 **文献标识码:** A **文章编号:** 2096-7675(2022)04-0412-09

引文格式: 彭湃, 冯新龙. 基于两重网格的深度学习方法求解定常偏微分方程[J]. 新疆大学学报(自然科学版)(中英文), 2022, 39(4): 412-420.

英文引文格式: PENG Pai, FENG Xinlong. Deep learning method based on two-grid for solving stationary partial differential equations[J]. Journal of Xinjiang University(Natural Science Edition in Chinese and English), 2022, 39(4): 412-420.

Deep Learning Method Based on Two-Grid for Solving Stationary Partial Differential Equations

PENG Pai, FENG Xinlong

(School of Mathematics and System Sciences, Xinjiang University, Urumqi Xinjiang 830017, China)

Abstract: With the progress of machine learning in many fields, physics-informed neural networks provide new ideas for solving partial differential equations, but this method is difficult to obtain high-precision numerical solutions. Absorbing the philosophy of physics-informed neural network and the two-grid solution of partial differential equations, this paper puts forward the deep learning method based on two-grid for solving stationary partial differential equations. For the neural network to solve the multi-objective problem, the dynamic weight strategy is adopted to balance the numerical difference between the items in the loss function, and alleviate the gradient ill-conditioned phenomenon. Finally, this paper gives several numerical experiments to verify the effectiveness of the deep learning method combined with dynamic weight strategy in improving the calculation accuracy.

Key words: stationary partial differential equations; deep learning; dynamic weight; high precision; two-grid

0 引言

随着深度学习研究的不断深入, 在不同的学科中产生了深刻变革, 如图像识别^[1]、认知科学^[2]等. 深度学习在处理复杂的系统模型中的预测任务时, 如何赋予深度学习这一强大函数近似器以先验知识成为学者的研究重点. 为了解决这个问题, 有些学者致力于设计专门的神经网络将给定的先验知识隐式嵌入到神经网络结构之中, 如利用卷积神经网络处理计算机视觉^[3]. 有些学者旨在通过适当的惩罚神经网络的损失函数赋予神经网络先验知识, 如利用物理信息神经网络^[4](Physics-Informed Neural Networks, PINN)求解偏微分方程^[5].

2019年, Raissi等人提出了物理信息神经网络用于求解偏微分方程的正问题及反问题. 随后诸多学者根据不同的方程及其应用背景对PINN进行探索. 在应用领域, Jin等建立了速度-压力、速度-涡度形式的NSFnets用于

* 收稿日期: 2021-09-22

基金项目: 新疆维吾尔自治区重点实验室开放课题(2020D04002).

作者简介: 彭湃(1997-), 男, 硕士生, 从事机器学习求解偏微分方程的研究, E-mail: penguinmath@163.com.

[†] 通讯作者: 冯新龙, 男, 教授, 主要从事偏微分方程数值解的研究, E-mail: fxlmath@xju.edu.cn.

求解层流和槽道湍流问题^[6]. 此外,一些学者将传统方法求解偏微分方程的技巧融入到PINN之中:如结合有限体积法在损失函数中增加界面约束做为正则化因子的DPINN^[7],Kharazmi等基于Petrov-Galerkin变分原理提出了VPINN求解偏微分方程^[8].通过分析前人的研究,不难发现PINN在求解偏微分方程时难以获得高精度解.为了改善神经网络求解偏微分方程的计算精度,本文提出了修正物理信息神经网络(Rectified Physics-Informed Neural Networks, RPINN)的解决方案.将两重网格^[9]求解方程的思想融入到PINN求解PDE的框架中,将PINN方法得到数值解及其梯度传递到新的神经网络之中,并建立相应的损失函数,通过训练新的网络从而得到PINN数值解的修正量.

1 预备知识

在这一部分,首先介绍前馈神经网络以及物理信息神经网络求解偏微分方程的主要思想,并详细阐述了神经网络求解多目标优化问题时采取的动态权重策略.

1.1 物理信息神经网络

在数学意义上,前馈神经网络可以表示为一个复杂的非线性函数 $f(x;\theta)_L$,其中 L 表示神经网络的层数, x 表示神经网络的输入向量, θ 表示神经网络的参数,包含神经网络的权重和偏置.对于一个全连接的前馈神经网络,可将其表示为(1)式:

$$f(x;\theta)_L = W_L(\cdots\sigma(W_1\sigma(W_0x+b_0)+b_1))+b_L \quad (1)$$

其中: $x \in \mathbb{R}^n$ 为输入向量, W_i 表示权重矩阵, b_i 为偏置向量, $\sigma(\cdot)$ 表示神经网络的激活函数.在神经网络中,激活函数的选取保证了网络的非线性特性.在本文中,选取双曲正切函数 $\sigma(\cdot) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ 作为神经网络的激活函数.

物理信息神经网络旨在求解出一个连续的数值解 $u(x)$ 使之满足方程和边界条件,为此本文定义定常偏微分方程如(2)式所示:

$$\begin{aligned} N_x[u] &= 0, \quad \text{在}\Omega\text{内} \\ u(x) &= g(x), \quad \text{在}\partial\Omega\text{上} \end{aligned} \quad (2)$$

其中: Ω 表示偏微分方程的求解区域, $\partial\Omega$ 为 Ω 的边界, $x \in \mathbb{R}^n$ 表示空间向量, N_x 表示非线性微分算子, u 表示方程满足的数值解, $g(x)$ 表示方程在边界处满足的函数.

将神经网络表示的非线性函数 $f(x;\theta)_L$ 代入到偏微分方程(2)式中,使之成为满足方程的一个潜在解.该潜在解满足残差 $r_\theta(x)$:

$$r_\theta(x) = N_x[f(x;\theta)_L] \quad (3)$$

同时在边界处满足:

$$f(x;\theta)_L(x) = g(x) \quad (4)$$

非线性函数 $f(x;\theta)_L$ 关于空间向量 x 的梯度通过自动微分^[10]求得.

若非线性函数 $f(x;\theta)_L$ 满足偏微分方程,则残差 $r_\theta(x) = 0$,并且有(4)式成立.根据方程的残差和边界条件建立神经网络的损失函数:

$$\mathcal{L}(x;\theta) = \lambda_f \mathcal{L}_f(x;\theta) + \lambda_b \mathcal{L}_b(x;\theta) \quad (5)$$

其中:

$$\mathcal{L}_f(x;\theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} \|r_\theta(x_f^i)\|_2^2 \quad (6)$$

$$\mathcal{L}_b(x;\theta) = \frac{1}{N_b} \sum_{i=1}^{N_b} \|f(x;\theta)_L(x_b^i) - g(x_b^i)\|_2^2 \quad (7)$$

其中: \mathcal{L}_f 和 \mathcal{L}_b 分别表示满足方程和边界的损失函数. λ_f 和 λ_b 分别表示损失函数 \mathcal{L}_f 和 \mathcal{L}_b 之间的权重因子,在优化过程中有效避免了梯度病态现象. $\{x_f^i, 0\}$ 表示满足残差 $r_\theta(x)$ 上的训练集,维数为 N_f , $\{x_b^i, g(x_b^i)\}$ 为满足边界的训练集,维数为 N_b .

当为神经网络建立损失函数 $\mathcal{L}(x; \theta)$ 后, 可通过梯度下降法、Adam算法、L-BFGS-B等一系列优化算法优化损失函数 $\mathcal{L}(x; \theta)$, 从而得到一组最优的神经网络参数 $\theta^* = \{W^*, b^*\}$. 最后将物理信息神经网络求解得到的数值解表示为 $u^*(x) = f(x; \theta^*)_L$. 其具体求解流程如图1所示.

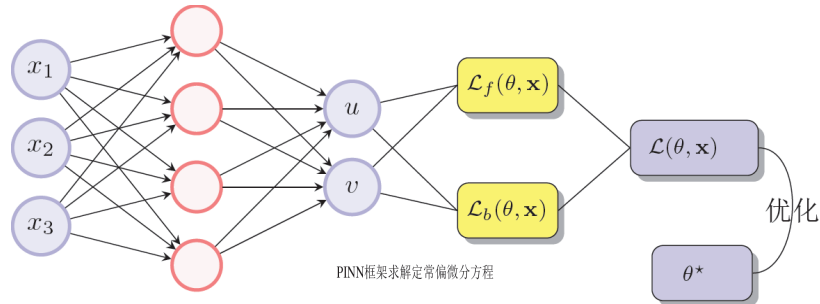


图 1 物理信息神经网络求解定常偏微分方程结构示意图

1.2 动态权重策略

在神经网络的训练过程中, 损失函数 $\mathcal{L}_f(x; \theta)$ 和 $\mathcal{L}_b(x; \theta)$ 数值差异较大, 在进行反向传播过程中, 往往侧重优化 $\mathcal{L}_f(x; \theta)$, 忽略在边界上的损失 $\mathcal{L}_b(x; \theta)$, 从而导致梯度病态现象. 为了缓解这一现象, 本文对神经网络的损失函数采取动态权重策略. 通过计算损失函数中 $\mathcal{L}_f(x; \theta)$ 和 $\mathcal{L}_b(x; \theta)$ 关于神经网络参数 θ 的梯度来动态调整权重 λ_b , 从而达到平衡损失函数 $\mathcal{L}_f(x; \theta)$ 和 $\mathcal{L}_b(x; \theta)$ 的作用.

考虑动态权重策略算法^[11]如下: 首先确定损失函数 \mathcal{L}_f 的权重 $\lambda_f = c$, 其中 c 为常数. 当采用梯度下降算法优化损失函数

$$\mathcal{L}(x; \theta) = \lambda_f \mathcal{L}_f(x; \theta) + \lambda_b \mathcal{L}_b(x; \theta) \quad (8)$$

时, 神经网络参数 θ 的更新公式为:

$$\theta_{k+1} = \theta_k - \eta \lambda_f \nabla_{\theta} \mathcal{L}_f - \eta \lambda_b \nabla_{\theta} \mathcal{L}_b \quad (9)$$

其中: θ_k 表示在第 k 次迭代过程中神经网络的参数, 超参数 η 表示神经网络的学习率, $\nabla(\cdot)$ 表示梯度算子, $\nabla_{\theta} \mathcal{L}_f$ 表示损失函数 $\mathcal{L}_f(x; \theta)$ 关于参数 θ_k 的梯度值. 然后计算中间变量

$$\hat{\lambda}_b = \frac{\max\{|\nabla_{\theta} \mathcal{L}_f(\theta_k)|\}}{|\nabla_{\theta} \mathcal{L}_b(\theta_k)|} \quad (10)$$

其中: $|\nabla_{\theta} \mathcal{L}_b(\theta_k)|$ 表示损失函数 $\mathcal{L}_b(x; \theta)$ 关于参数 θ_k 求梯度所得梯度值的平均值, $\max\{|\nabla_{\theta} \mathcal{L}_f(\theta_k)|\}$ 表示损失函数 $\mathcal{L}_f(x; \theta)$ 关于参数 θ_k 求梯度所得梯度值的最大值. 然后根据中间变量 $\hat{\lambda}_b$ 和第 k 次迭代过程中 $\mathcal{L}_b(x; \theta)$ 的权重值 $\lambda_b^{(k)}$ 计算得到第 $k+1$ 次迭代中的权重值 $\lambda_b^{(k+1)}$:

$$\lambda_b^{(k+1)} = (1 - \mu) \lambda_b^{(k)} + \mu \hat{\lambda}_b \quad (11)$$

其中: μ 是一个取值范围为(0, 1)的超参数. 同时定义未采取动态权重策略为(D0)策略.

2 基于两重网格的深度学习算法

两重网格算法最初来源于对方程的迭代求解, 它将细网格方程限制到粗网格上进行求解, 然后将所得的解延拓到细网格之上, 最后与原来的解组合形成细网格上的精确解. 本文提出的基于两重网格的深度学习算法来源于两重网格求解方程的思想, 将PINN得到的数值解及其梯度传递到新的神经网络之中, 通过为新的神经网络建立损失函数以此获得数值解的修正量.

首先, 将PINN求解偏微分方程得到的数值解 $u^*(x) = f(x; \theta^*)_L$ 代入到偏微分方程(2)式中, 可得(12)式:

$$\begin{aligned} N_x[u^*] &\approx 0, \quad \text{在}\Omega\text{内} \\ u^*(x) &\approx g(x), \quad \text{在}\partial\Omega\text{上} \end{aligned} \quad (12)$$

(12)式表明由PINN计算得到的数值解 $u^*(x)$ 与真解 u 存在误差 $\varepsilon = u - u^*(x)$, 接下来为误差 ε 建立新的前馈神经网络 $\varepsilon(x; \theta)_L$, 使之满足(13)式:

$$\begin{aligned} N_x[u^* + \varepsilon(x; \theta)_L] &= 0, & \text{在}\Omega\text{内} \\ u^*(x) + \varepsilon(x; \theta)_L &= g(x), & \text{在}\partial\Omega\text{上} \end{aligned} \tag{13}$$

根据(13)式, 构建神经网络 $\varepsilon(x; \theta)_L$ 与PINN数值解之间的残差方程 $r_\theta^*(x; t)$:

$$r_\theta^*(x; t) = N_x[\varepsilon(x; \theta)_L + u^*] \tag{14}$$

并根据(13)和(14)式, 为神经网络 $\varepsilon(x; \theta)_L$ 建立损失函数

$$\mathcal{L}^\varepsilon(x; \theta) = \lambda_f^\varepsilon \mathcal{L}_f^\varepsilon(x; \theta) + \lambda_b^\varepsilon \mathcal{L}_b^\varepsilon(x; \theta) \tag{15}$$

其中:

$$\mathcal{L}_f^\varepsilon(x; \theta) = \frac{1}{N_f^\varepsilon} \sum_{i=1}^{N_f^\varepsilon} \|r_\theta^*(x_f^i; t_f^i)\|_2^2 \tag{16}$$

$$\mathcal{L}_b^\varepsilon(x; \theta) = \frac{1}{N_b^\varepsilon} \sum_{i=1}^{N_b^\varepsilon} \|\varepsilon(x; \theta)_L(x_b^i) - u^*(x_b^i) - g(x_b^i)\|_2^2 \tag{17}$$

λ_f^ε 和 λ_b^ε 为权重因子. $\{x_f^i, 0\}$ 表示神经网络 $\varepsilon(x; \theta)_L$ 满足残差 $r_\theta^*(x)$ 的训练集, 为提高计算精度, 训练集的选取与神经网络 $f(x; \theta)_L$ 的训练集相互独立, $\{x_b^i, g(x_b^i)\}$ 表示神经网络 $\varepsilon(x; \theta)_L$ 在边界处的训练集. 在神经网络 $\varepsilon(x; \theta)_L$ 的损失函数 $\mathcal{L}^\varepsilon(x; \theta)$ 中, $u^*(x_f^i)$ 和 $u^*(x_b^i)$ 为神经网络 $f(x; \theta)_L$ 传递到神经网络 $\varepsilon(x; \theta)_L$ 的数值解, 数值解 u^* 关于向量 x 的梯度和二阶导数通过自动微分求解.

在本文中, 采用Adam算法优化神经网络的损失函数 $\varepsilon(x; \theta)_L$, 同时为了有效获得神经网络的最优参数, 采取学习率衰减策略, 即经过迭代步长 N 次迭代后, 学习率 η 衰减为原来的 ρ 倍. 通过对神经网络不断优化, 得到满足数值解 $u^*(x)$ 的修正量 $\varepsilon^* = \varepsilon(x; \theta)_L$. 从而将RPINN求解偏微分方程的数值解表示为: $u_R = u^*(x) + \varepsilon(x; \theta)_L$, 具体求解流程如图2所示.

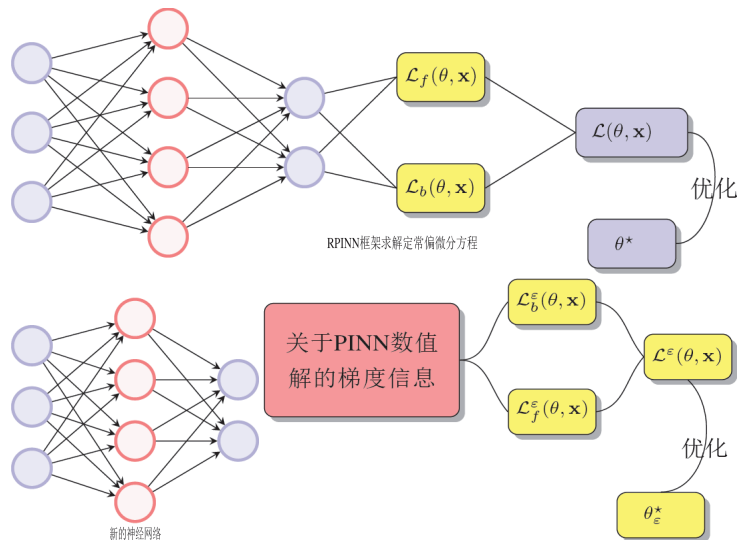


图 2 修正物理信息神经网络求解定常偏微分方程结构示意图

3 数值算例

为了验证结合动态权重策略的深度学习方法在提高计算精度上的有效性, 本文给出了若干数值实验, 分别为对流扩散反应方程和三维Navier-Stokes方程的求解, 并且定义相对 L_2 误差为 $\epsilon_V = \frac{\|\tilde{V} - V\|_2}{\|V\|_2}$.

例 1 对流扩散反应方程属于流体力学中的基本方程. 大气污染、河流污染等诸多变化过程都可以用对流扩散反应方程来表示. 对于求解此类方程, 常用的求解方法有有限元法、多重网格法. 下面本文通过求解一个二维的对流扩散反应方程来验证基于两重网格的深度学习算法的有效性. 定义对流扩散反应方程如下所示^[12]

$$\begin{aligned} -\alpha\Delta u + \beta\nabla u + u &= f, \quad \text{在}\Omega\text{内} \\ u &= 0, \quad \text{在}\partial\Omega\text{上} \end{aligned} \quad (18)$$

其中: $\alpha = 0.1$ 是扩散项参数, $\beta = (2, 3)$, f 是满足方程的源项, 求解区域为 $\Omega = [0, 1]^2$. 定义方程满足的真解为:

$$u(x, y) = 2 \sin x \cdot (1 - e^{-\frac{2(1-x)}{\alpha}}) \cdot y^2 \cdot (1 - e^{-\frac{y-1}{\alpha}}) \quad (19)$$

在PINN求解对流扩散反应方程时, 可根据方程满足的残差及其边界建立神经网络的损失函数:

$$\mathcal{L}(x; \theta) = \lambda_f \mathcal{L}_f(x; \theta) + \lambda_b \mathcal{L}_b(x; \theta) \quad (20)$$

其中:

$$\mathcal{L}_f(x; \theta) = \frac{1}{N_f} \sum_{i=1}^{N_f} \|- \alpha \Delta u_i + \beta \nabla u_i + u_i - f_i \|_2^2 \quad (21)$$

$$\mathcal{L}_b(x; \theta) = \frac{1}{N_b} \sum_{i=1}^{N_b} \|u_i\|_2^2 \quad (22)$$

N_f, N_b 分别为在求解区域 Ω 以及边界上选取训练点的个数.

在本算例中, PINN方法中神经网络 $f(x; \theta)_L$ 的参数设置如下: 神经网络的网络结构为 $2 + [50] \times 4 + 1$, 即输入层含有2个输入神经元, 中间隐藏层的个数为4, 每层包含有50个神经元, 输出层含有1个输出神经元. 训练集的维数为: $N_f = 10\ 500, N_b = 2\ 800$. 初始学习率 η 设置为 $\eta = 10^{-3}$, 衰减步长为 $N = 1\ 000$, 学习率的衰减率为 $\rho = 0.9$. 动态权重策略中采取的超参数为 $\mu = 0.5, \lambda_f = 1$, 采取Adam优化算法的迭代次数为8 000, 并采用L-BFGS-B算法加速收敛.

在RPINN方法中神经网络 $f(x; \theta)_L$ 的参数设置如下: 神经网络的网络结构为 $2 + [50] \times 4 + 1$, 训练集的维数为: $N_f = 8\ 000, N_b = 2\ 000$. 初始学习率 η 设置为 $\eta = 10^{-3}$, 衰减步长为 $N = 1\ 000$, 学习率的衰减率为 $\rho = 0.9$. 动态权重策略中采取的超参数为 $\mu = 0.5, \lambda_f = 1$, 采取Adam优化算法的迭代次数为3 000. 并将PINN方法训练得到满足偏微分方程的数值解记作 \hat{u} .

将数值解 \hat{u} 代入(18)式, 并根据数值解及其梯度构建神经网络 $\varepsilon(x; \theta)_L$ 的损失函数:

$$\mathcal{L}^\varepsilon(x; \theta) = \lambda_f^\varepsilon \mathcal{L}_f^\varepsilon(x; \theta) + \lambda_b^\varepsilon \mathcal{L}_b^\varepsilon(x; \theta) \quad (23)$$

其中:

$$\mathcal{L}_f^\varepsilon(x; \theta) = \frac{1}{N_f^\varepsilon} \sum_{i=1}^{N_f^\varepsilon} \|- \alpha \Delta \varepsilon_i + \beta \nabla \varepsilon_i + \varepsilon_i - \hat{f}_i \|_2^2 \quad (24)$$

$$\mathcal{L}_b^\varepsilon(x; \theta) = \frac{1}{N_b^\varepsilon} \sum_{i=1}^{N_b^\varepsilon} \|\varepsilon_i + \hat{u}_i\|_2^2 \quad (25)$$

\hat{f}_i 的表达式为: $\hat{f}_i = \alpha \Delta \hat{u}_i - \beta \nabla \hat{u}_i - \hat{u}_i + f_i$.

神经网络 $\varepsilon(x; \theta)_L$ 的参数设置如下: 神经网络的网络结构为 $2 + [50] \times 4 + 1$, 训练集的维数为: $N_f^\varepsilon = 2\ 500, N_b^\varepsilon = 800$. 初始学习率 η 设置为 $\eta = 10^{-3}$, 学习率的衰减步长为 $N = 1\ 000$, 学习率的衰减率为 $\rho = 0.9$. 动态权重策略中采取的超参数为 $\mu = 0.5, \lambda_f^\varepsilon = 1$, 采取Adam优化算法进行15 000次迭代, 并采取L-BFGS-B算法加速网络的收敛. 最后将神经网络 $\varepsilon(x; \theta)_L$ 得到的修正量记作 $\varepsilon^* = \varepsilon(x; \theta)_L$, 从而RPINN方法得到数值解为 $u = \hat{u} + \varepsilon^*$.

下面本文将多重网格、PINN、RPINN方法得到的数值解与真解进行对比分析. 值得说明的是: 在多重网格求解方程时, 迭代方法采用SOR迭代, PINN方法选取训练集的维数为13 300, 具体实验数据如表1所示. 通过分析可知: 与多重网格相比, RPINN方法利用更少的数据获得高精度的数值解. 同时, 采取动态权重策略(D1)能够有效缓解梯度病态现象, 得到高精度的数值解. RPINN与PINN方法相比, 计算精度提高了3至8倍. 关于RPINN(D1)、PINN(D1)、多重网格求解得到的数值解与真解的图像如图3所示.

表 1 多重网格、PINN和RPINN求解对流扩散反应方程之间的数据对比

方法	多重网格	PINN(D0)	PINN(D1)	RPINN(D0)	RPINN(D1)
时间/s	109.32	830.56	901.82	897.78	971.14
数据集点数	262 144	13 300	13 300	13 300	13 300
绝对误差	0.002 017 82	0.001 017 55	0.000 817 85	0.000 384 11	0.000 113 89
相对 L_2 误差	0.013 352 03	0.006 365 21	0.004 276 90	0.002 183 00	0.000 746 97

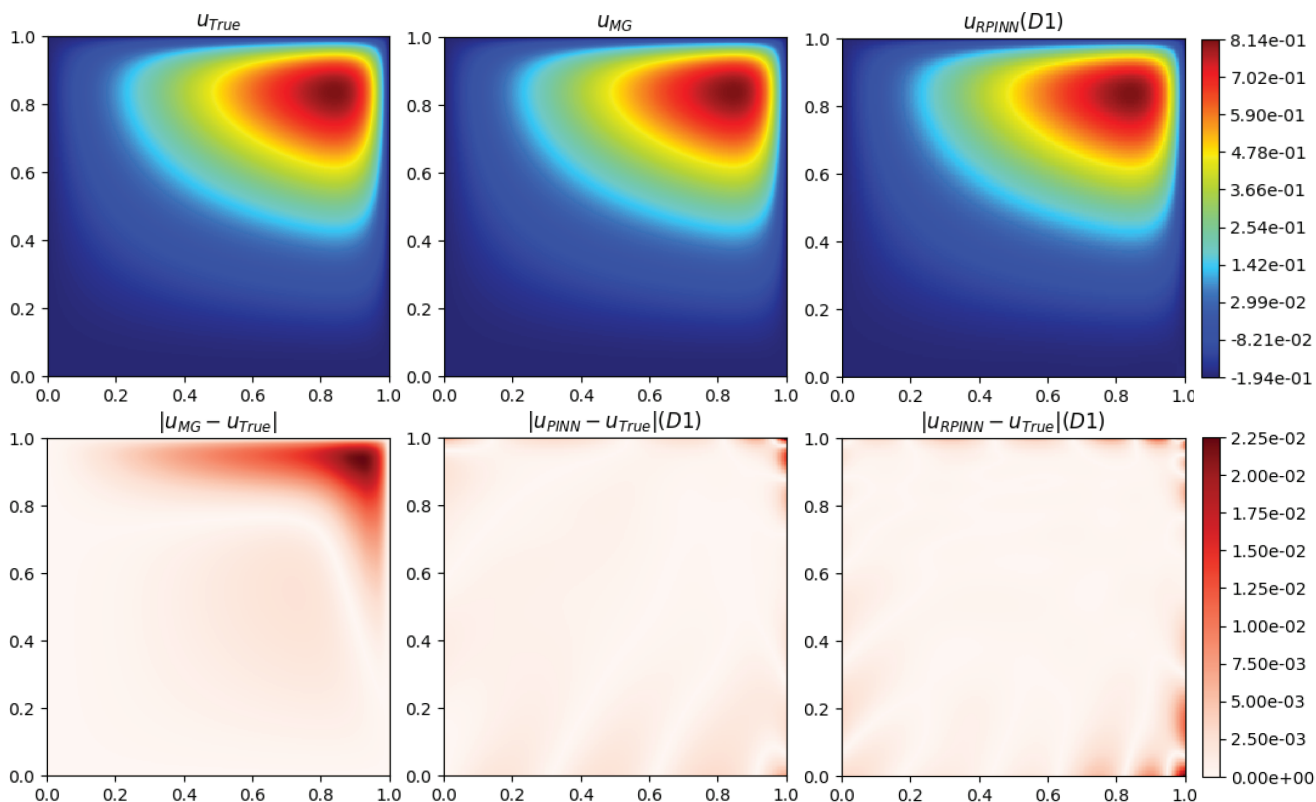


图 3 多重网格、PINN、RPINN方法之间的误差比较

然后, 本文给出了RPINN求解对流扩散反应方程时, 采取动态权重策略(D0)、(D1)的RPINN关于神经网络参数 θ 的梯度 $\nabla_{\theta} \mathcal{L}_f^{\varepsilon}(\theta_k)$ 和 $\nabla_{\theta} \mathcal{L}_b^{\varepsilon}(\theta_k)$ 的直方分布图, 分别如图4和图5所示. 结果表明结合动态权重策略的RPINN中 $\nabla_{\theta} \mathcal{L}_f^{\varepsilon}(\theta_k)$ 和 $\nabla_{\theta} \mathcal{L}_b^{\varepsilon}(\theta_k)$ 的值均趋向于0, 并且 $\nabla_{\theta} \mathcal{L}_f^{\varepsilon}(\theta_k)$ 和 $\nabla_{\theta} \mathcal{L}_b^{\varepsilon}(\theta_k)$ 数值之间差异小, 验证了动态权重策略能有效缓解梯度病态现象.

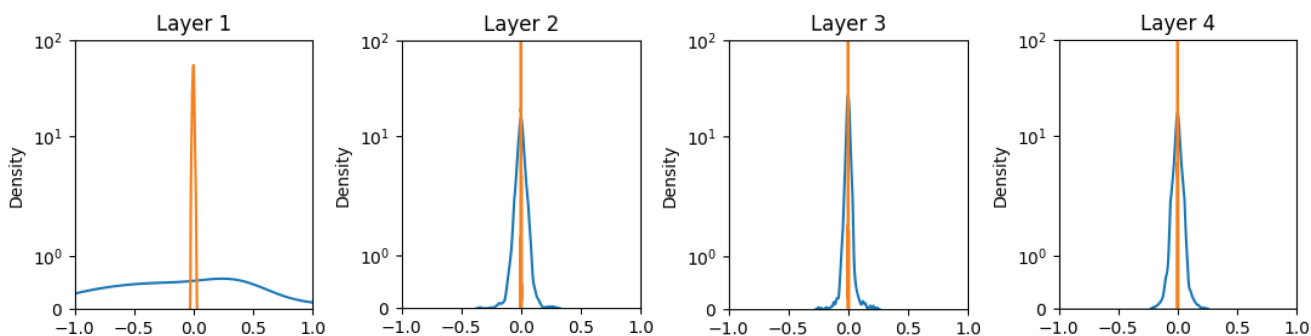


图 4 采用动态权重策略D0的RPINN中 $\nabla_{\theta} \mathcal{L}_f^{\varepsilon}(\theta_k)$ 和 $\nabla_{\theta} \mathcal{L}_b^{\varepsilon}(\theta_k)$ 梯度值的直方分布图

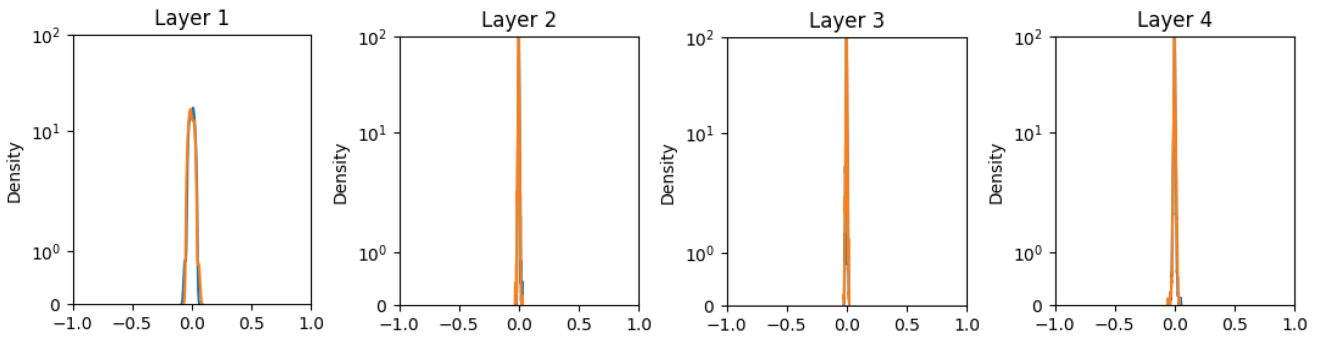


图 5 采用动态权重策略D1的RPINN中 $\nabla_{\theta} \mathcal{L}_f^{\varepsilon}(\theta_k)$ 和 $\nabla_{\theta} \mathcal{L}_b^{\varepsilon}(\theta_k)$ 梯度值的直方分布图

最后, 本文分析了RPINN的神经网络结构对修正量 ε^* 训练结果的影响. 首先以神经网络结构为 $2 + [50] \times 4 + 1$ 的PINN训练得到的数值解作为基准, 通过改变第二个神经网络的结构对同一数值解进行修正. 具体的神经网络结构设计和误差如表2所示. 结果表明: 随着神经网络的网络层数和每层神经元个数的增加, 对修正量 ε^* 训练效果越好.

表 2 不同网络结构对数值精度影响的比较

网络结构	绝对误差	网络结构	绝对误差
$2 + [20] \times 1 + 1$	0.006 246 14	$2 + [50] \times 1 + 1$	0.006 395 65
$2 + [20] \times 2 + 1$	0.006 384 02	$2 + [50] \times 2 + 1$	0.005 200 51
$2 + [20] \times 3 + 1$	0.003 271 78	$2 + [50] \times 3 + 1$	0.000 968 43
$2 + [20] \times 4 + 1$	0.000 576 13	$2 + [50] \times 4 + 1$	0.000 344 40
$2 + [20] \times 5 + 1$	0.000 127 93	$2 + [50] \times 5 + 1$	0.000 101 14

例 2 Navier-Stokes方程是一类描述流体流动的非线性方程, 在流体力学中占有重要地位. 本算例中, 定义稳态不可压缩的Navier-Stokes方程如(26)式所示:

$$\begin{aligned}
 -\frac{1}{Re} \nabla^2 u + (u \cdot \nabla) u + \nabla p &= f, \quad \text{在 } \Omega \text{ 内} \\
 \nabla \cdot u &= 0, \quad \text{在 } \Omega \text{ 内} \\
 u &= u_{\Gamma}, \quad \text{在 } \partial\Omega \text{ 上}
 \end{aligned}
 \tag{26}$$

其中: $u = [u, v, w]^T$ 是无量纲的速度向量, p 是无量纲压力, 雷诺数 $Re = UL/v$, 其中 U 和 L 分别是特征速度和长度, v 是流体的运动粘度, $f = [f_1, f_2, f_3]$ 为满足方程的源项, Γ 表示狄利克雷边界条件. 求解区域 $\Omega = [-1, 1]^3$, 定义方程满足的真解为:

$$\begin{aligned}
 u(x, y, z) &= -a[e^{ax} \sin(ay + dz) + e^{az} \cos(ax + dy)] \\
 v(x, y, z) &= -a[e^{ay} \sin(az + dx) + e^{ax} \cos(ay + dz)] \\
 w(x, y, z) &= -a[e^{az} \sin(ax + dy) + e^{ay} \cos(az + dx)] \\
 p(x, y, z) &= -\frac{1}{2} a^2 [e^{2ax} + e^{2ay} + e^{2az} + 2 \sin(ax + dy) \cos(az + dx) e^{a(y+z)} \\
 &\quad + 2 \sin(ay + dz) \cos(ax + dy) e^{a(z+x)} + 2 \sin(az + dx) \cos(ay + dz) e^{a(x+y)}]
 \end{aligned}
 \tag{27}$$

其中: $a = d = 1$.

在本算例中, PINN方法中神经网络参数设置如下: 神经网络的网络结构为 $3 + [50] \times 4 + 4$, 即输入层包含3个神经元, 中间隐藏层的个数为4, 每层包含50个神经元, 输出层包含4个神经元. 训练集的维数为: $N_f = 11\ 375$, $N_b = 4\ 200$. 初始学习率 η 设置为 $\eta = 10^{-3}$, 衰减步长为 $N = 1\ 000$, 学习率的衰减率为 $\rho = 0.9$. 在动态权重策略中采取的超参数为 $\mu = 0.5$, $\lambda_f = 1$, Adam算法的迭代次数为8 000, 采用L-BFGS-B算法加速收敛.

在RPINN方法中神经网络 $f(x; \theta^*)_L$ 的参数设置如下: 神经网络的网络结构为 $3 + [50] \times 4 + 4$, 训练集的维数为: $N_f = 8\ 000$, $N_b = 3\ 000$. 初始学习率 η 设置为 $\eta = 10^{-3}$, 衰减步长为 $N = 1\ 000$, 学习率衰减率为 $\rho = 0.9$. 在动态权重策略中采取的超参数为 $\mu = 0.5$, $\lambda_f = 1$, Adam 算法的迭代次数为4 000.

神经网络 $\varepsilon(x; \theta)_L$ 的参数设置如下: 神经网络的网络结构为 $3 + [40] \times 4 + 4$, 训练集的维数为: $N_f^\varepsilon = 3\ 375$, $N_b^\varepsilon = 1\ 200$. 初始学习率 η 设置为 $\eta = 10^{-3}$, 衰减步长为 $N = 1\ 000$, 学习率的衰减率为 $\rho = 0.9$. 动态权重策略中采取的超参数为 $\mu = 0.5$, $\lambda_f^\varepsilon = 1$, Adam算法的迭代次数为12 000, 采用L-BFGS-B算法加速收敛. 经过对神经网络 $\varepsilon(x; \theta)_L$ 不断训练, 最终得到数值解 \hat{u} 的修正量为 $\varepsilon^* = \varepsilon(x; \theta)_L$, 从而将RPINN方法得到数值解记为 $u = \hat{u} + \varepsilon^*$.

最后比较了PINN、RPINN得到数值解在 $z = 0$ 平面上与真解的误差, 如图6所示. 实验结果表明, PINN方法和RPINN方法在求解偏微分方程时, 与求解区域内部相比, 在边界上的误差偏大. 有关速度场的数值解与真解的误差如表3所示. 通过比较分析可知, RPINN方法与PINN方法相比, 能够有效提高数值精度.

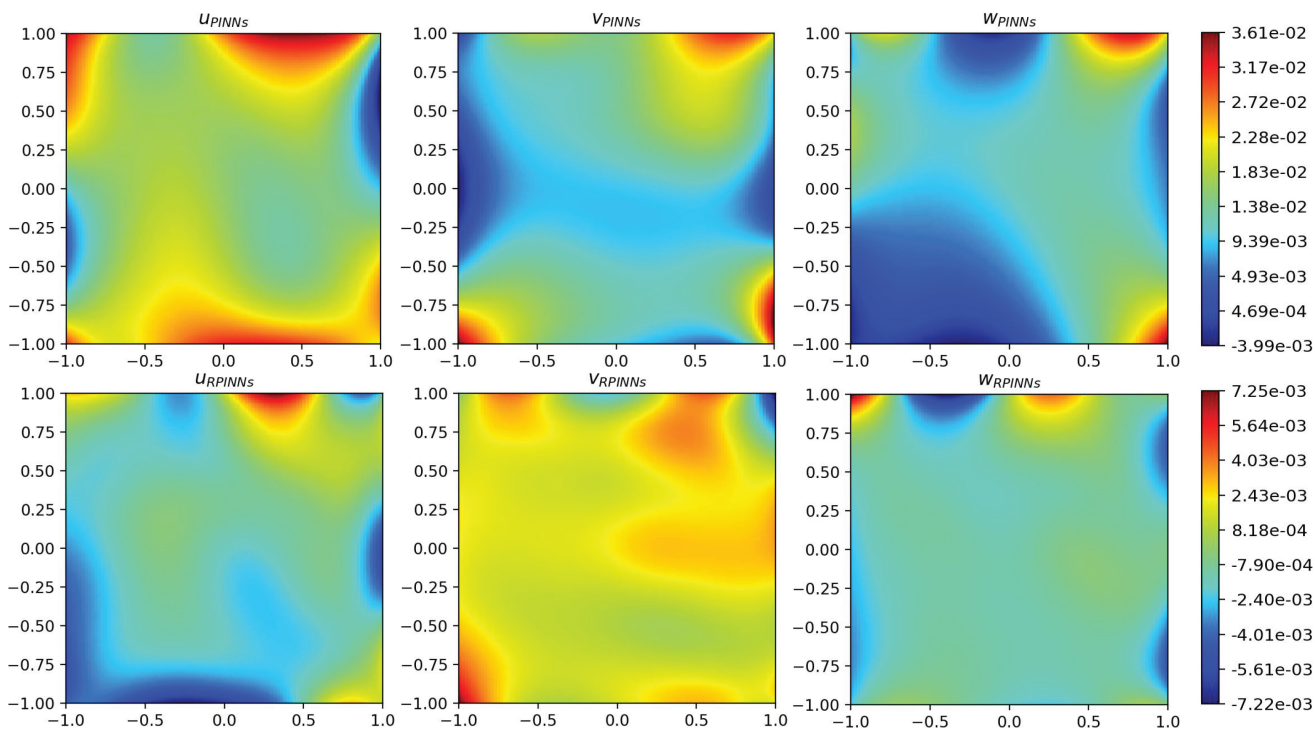


图 6 PINN、RPINN数值解在 $z = 0$ 平面上与真解的误差图

表 3 PINN、RPINN方法求解Navier-Stokes方程的数值比较

	PINN方法			RPINN方法		
	速度 u	速度 v	速度 w	速度 u	速度 v	速度 w
L_1 误差	0.000 828 25	0.000 898 38	0.000 808 03	0.000 529 62	0.000 615 80	0.000 560 91
L_2 误差	0.000 813 26	0.000 849 00	0.000 803 80	0.000 610 72	0.000 622 84	0.000 562 00

4 结论

本文根据两重网格求解偏微分方程的思想, 对物理信息神经网络进行改进, 提出了基于两重网格的深度学习方法求解定常偏微分方程. 然后针对优化过程中出现的梯度病态现象, 采取了动态权重策略. 最后通过若干数值实验验证, 本文提出的基于两重网格的深度学习方法与原始的物理信息神经网络相比, 在计算精度上提高了3至8倍. 关于如何设计出快速稳定的深度学习方法以减少计算时间以及进一步提高数值精度将是以后研究工作的重点.

参考文献:

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.

- [2] LAKE B M, SALAKHUTDINOV R, TENENBAUM J B. Human-level concept learning through probabilistic program induction[J]. *Science*, 2015, 350: 1332-1338.
- [3] LECUN Y, BENGIO Y. Convolutional networks for images, speech, and time series[J]. *The Handbook of Brain Theory and Neural Networks*, 1995, 3361(10): 1995.
- [4] RAISSI M, PERDIKARIS P, KARNIADAKIS G E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. *Journal of Computational Physics*, 2019, 378: 686-707.
- [5] TUKEZI A, ABUDUREXITI A. The study of a new numerical method for parabolic partial differential equations[J]. *Journal of Xinjiang University(Natural Science Edition)*, 2014, 31(1): 64-69.
- [6] JIN X W, CAI S Z, LI H, et al. NSFnets(Navier-Stokes Flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations[J]. *Journal of Computational Physics*, 2021, 426: 109951.
- [7] DWIVEDI V, PARASHAR N, SRINIVASAN B. Distributed physics informed neural network for data-efficient solution to partial differential equations[J]. *arXiv preprint arXiv*, 2019, <https://doi.org/10.48550/arXiv.1907.08967>.
- [8] KHARAZMI E, ZHANG Z Q, KARNIADAKIS G E. Variational physics-informed neural networks for solving partial differential equations[J]. *arXiv preprint arXiv*, 2019, <https://doi.org/10.48550/arXiv.1912.00873>.
- [9] WANG X J, JIA H E. A two-grid method with backtracking for the incompressible Brinkman-Forchheimer equations[J]. *Journal of Xinjiang University(Natural Science Edition in Chinese and English)*, 2021, 38(3): 275-284.
- [10] BAYDIN A G, PEARLMUTTER B A, RADUL A A, et al. Automatic differentiation in machine learning: a survey[J]. *The Journal of Machine Learning Research*, 2018, 18: 1-43.
- [11] WANG S F, TENG Y J, PERDIKARIS P. Understanding and mitigating gradient pathologies in physics-informed neural networks[J]. *SIAM Journal on Science Computing*, 2021, 43(5): 3055-3081.
- [12] WU J L, GUI D W, LIU D M, et al. The characteristic variational multiscale method for time dependent conduction-convection problems[J]. *International Communications in Heat and Mass Transfer*, 2015, 68: 58-68.

责任编辑: 赵新科

(上接第 411 页)

- [7] RIBEIRO R, BONHOEFFER S. Production of resistant HIV mutants during antiretroviral therapy[J]. *Proceedings of the National Academy of Sciences*, 2000, 97(14): 7681-7686.
- [8] RONG L, FENG Z, PERELSON A. Emergence of HIV-1 drug resistance during antiretroviral treatment[J]. *Bulletin Mathematical Biology*, 2007, 69(6): 2027-2060.
- [9] 陈伟, 滕志东. 敏感型和耐药型HIV 毒株感染模型的全局动力学[J]. *四川师范大学学报(自然科学版)*, 2021, 44(5): 615-625.
- [10] HUANG G, MA W, TAKEUCHI Y. Global analysis for delay virus dynamics model with Beddington-DeAngelis functional response[J]. *Applied Mathematics Letters*, 2011, 24(7): 1199-1203.
- [11] CHEN W, TUERXUN N, TENG Z. The global dynamics in a wild-type and drug-resistant HIV infection model with saturated incidence[J]. *Advances in Difference Equations*, 2020, <https://doi.org/10.1186/s13662-020-2497-2>.
- [12] YANG Y, ZOU L, RUAN S. Global dynamics of a delayed within-host viral infection model with both virus-to-cell and cell-to-cell transmissions[J]. *Mathematical Biosciences*, 2015, 270: 183-191.
- [13] 曼合布拜·热合木, 李晓娜. 释放毒素的浮游植物与浮游动物相互作用时滞模型分析[J]. *新疆大学学报(自然科学版)*, 2019, 36(3): 303-311.
- [14] 艾合麦提·麦麦提阿吉. 含时滞非线性扩散合作系统的正周期解存在性与吸引性[J]. *新疆大学学报(自然科学版)*, 2019, 36(1): 1-10.
- [15] HALE J. *Theory of functional differential equations*[M]. New York-Heidelberg: Applied Mathematical Sciences, Springer-Verlag, 1977.
- [16] 马知恩, 周义仓. *传染病动力学的数学建模与研究*[M]. 北京: 科学出版社, 2004.
- [17] WANG W, MA Z. Harmless delays for uniform persistence[J]. *Journal of Mathematical Analysis and Applications*, 1991, 158(1): 256-268.

责任编辑: 赵新科