文章编号:1671-9352(2026)02-0026-11 DOI:10.6040/j.issn.1671-9352.0.2024.244

微分方程特征值问题的物理信息神经网络数值解法

唐瑜,袁利军*
(重庆工商大学数学与统计学院, 重庆 400067)

摘要:针对微分方程特征值问题,提出一个改进的物理信息神经网络求解方法和两阶段训练法。该方法可以求解多个绝对值最小特征值、求解离初始值最近的特征值问题和重特征值问题等。通过一维、二维正方形区域以及 L 形区域中拉普拉斯算子特征值问题的数值算例表明本文方法比现有方法精度更高。

关键词:物理信息神经网络;微分方程;特征值问题
中图分类号:O242 **文献标志码**:A
引用格式:唐瑜,袁利军. 微分方程特征值问题的物理信息神经网络数值解法[J]. 山东大学学报(理学版),2026,61(2):26-36.

Numerical solution of physically informed neural networks for eigenvalue problems of differential equations

TANG Yu, YUAN Lijun*
(School of Mathematics and Statistics, Chongqing Technology and Business University, Chongqing 400067, China)

Abstract: For eigenvalue problems of differential equations, an improved physical information neural network solution and a two-stage training method are proposed. The new method can solve multiple minimum eigenvalues, the eigenvalue problem closest to the initial value and the multiple eigenvalue problem. Numerical examples of Laplace operator eigenvalue problems in 1D and 2D square regions as well as L-shaped regions show that the new method is more accurate than the existing methods.

Key words: physically informed neural networks; differential equations; eigenvalue problems

0 引言

近年来,深度学习在动力学分析[1]、蛋白质作用关系预测[2]和图像处理[3]等多个领域取得了巨大的成功。随着深度学习的发展,利用深度神经网络求解偏微分方程的研究也越来越活跃[4]。相对于传统的偏微分方程数值求解算法,由于深度神经网络不需要划分网格、处理非线性项和随机项方便快捷,因此基于深度神经网络的偏微分方程数值求解算法在求解非线性问题、随机偏微分方程、反问题和高维偏微分方程等问题上具有巨大优势[5-6]。利用深度神经网络求解偏微分方程需要构造合适的损失函数。目前构造损失函数的方法主要有以下 4 种:鄂维南等[7]提出的深度里兹法(deep Ritz method), Sirignano 等[8]提出的深度伽辽金方法(deep Galerkin method), Raissi 等[9]提出的物理信息神经网络(physics-informed neural networks, PINN), 以及 Zang 等[10]提出的弱对抗神经网络法(weak adversarial network)。由于物理信息神经网络在构造损失函数上的便利性和易于与其他方法融合的特性,此方法被用于求解各种科学问题,例如纳维-斯托克斯方程、波动方程、KdV 方程、伯格斯方程等[11-16],以及用于求解一类偏微分方程,而不是某一个偏微分方程[17-18]。

收稿日期:2024-07-11; **网络出版时间**:2025-06-11
基金项目:重庆市自然科学基金面上项目(CSTB2022NSCQ-MSX0610)
第一作者:唐瑜(1999—),女,硕士研究生,研究方向为偏微分方程数值解. E-mail:384153030@qq.com
*通信作者:袁利军(1982—),男,教授,博士,研究方向为深度学习和科学计算. E-mail:ljyuan@ctbu.edu.cn

神经网络也可以用来求解偏微分方程特征值问题。鄂维南等在文献[7]中首次表明深度神经网络可以求解高维偏微分方程的特征值问题。给出的方法可以计算最小特征值和对应的特征函数。Han 等[19]通过将高维偏微分方程的特征值问题转化为一个后向偏微分方程问题,再利用深度后向随机微分方程法求解。Yang 等[20]基于求解代数方程特征值问题的幂方法提出了一个神经网络算法求解线性微分方程的特征值问题,此方法可以求解最小特征值、离给定初始值最近的特征值和最大特征值。Ben-Shaul 等[21]提出了一个求解微分算子前若干个最小特征值和对应特征函数的物理信息神经网络算法。目前,利用神经网络求解特征值非常大时的特征函数,以及求解离给定初始值最近的多个特征值比较困难。

本文通过改进 Ben-Shaul 等[21]的算法得到一个新的求解偏微分方程特征值问题的物理信息神经网络算法,该方法可计算前若干个绝对值最小特征值,也可计算离给定初始值最近的一个或者多个特征值问题,也可求解重特征值问题,且通过精度对比,本文算法要比 Ben-Shaul 等的算法精度更高。

1　PINN 求解特征值问题

1.1　特征值问题

考虑微分方程

$$\begin{cases}\mathscr{N}[u](X)=-\lambda u(X), & X\in\Omega,\\ u\mid_{\partial\Omega}=0\end{cases} \tag{1}$$

的特征值问题,其中,$\mathscr{N}[\cdot]$是一个线性或非线性微分算子,X 为空间坐标,u 为特征函数,λ 为特征值,Ω 表示为计算区域,$\partial\Omega$ 表示 Ω 区域的边界。已知特征函数 u 后特征值 λ 可由瑞利商计算得到

$$R(\tilde{u})=\frac{-\int_{\Omega}u\mathscr{N}(u)\mathrm{d}X}{\int_{\Omega}u^2\mathrm{d}X}, \tag{2}$$

特征函数乘以一个非零常数还是一个特征函数,在计算过程中对特征函数做如下归一化处理:

$$\int_{\Omega}u^2\mathrm{d}X=1。 \tag{3}$$

若算子 $\mathscr{N}[\cdot]$是自伴的,则属于不同特征值的特征函数,以及重特征值的特征函数之间是相互正交的,即

$$\int_{\Omega}u_iu_k\mathrm{d}X=0, \tag{4}$$

其中,u_i,u_k 为不同特征值的特征函数或属于重特征值的不同特征函数。

1.2　物理信息神经网络求解特征值问题

用神经网络来近似特征函数 $u(X)$,其中 X 表示神经网络的输入,输出记为 $\tilde{u}(X;\theta)$,θ 为神经网络的参数,即所有的权重和阈值。本文使用文献[22]中的多尺度神经网络结构求解大特征值和对应的特征函数。

1.2.1　求解单个特征值问题

给定初始特征值 λ_0,目的是求解最靠近此初始值的特征值和对应的特征函数。构造损失函数如下:

$$L(\theta)=L_{\mathrm{eq}}(\theta)+\alpha L_{\mathrm{bd}}(\theta)+\beta L_{\mathrm{unit}}(\theta)+\rho L_{\mathrm{lambd}}(\theta), \tag{5}$$

其中

$$L_{\mathrm{eq}}(\theta)=\frac{\frac{1}{N_f}\sum_{j=1}^{N_f}\left|\frac{\mathscr{N}(\tilde{u}(X_j^f;\theta))}{R(\tilde{u})}+\tilde{u}(X_j^f;\theta)\right|^2}{I(\tilde{u})},\quad L_{\mathrm{bd}}(\theta)=\frac{\frac{1}{N_b}\sum_{j=1}^{N_b}\tilde{u}^2(X_j^b;\theta)}{I(\tilde{u})},$$

$$L_{\mathrm{unit}}(\theta)=\left(1-\frac{1}{I(\tilde{u})}\right)^2,\quad L_{\mathrm{lambd}}(\theta)=\left(\frac{R(\tilde{u})}{\lambda_0}-1\right)^2,$$

$$R(\tilde{u})=\frac{-\frac{1}{N_f}\sum_{j=1}^{N_f}[\tilde{u}(X_j^f;\theta)\mathscr{N}(\tilde{u}(X_j^f;\theta))]}{I(\tilde{u})},\quad I(\tilde{u})\triangleq\frac{1}{N_f}\sum_{j=1}^{N_f}\tilde{u}^2(X_j^f;\theta),$$

α、β、ρ 为权重,L_{eq}为方程残差损失函数,L_{bd}为边界损失函数,L_{unit}为特征函数的归一化损失函数,L_{lambd}为特征值损失函数。$\{X_j^f\}_{j=1}^{j=N_f}$为区域 Ω 内部采样点,$\{X_j^b\}_{j=1}^{j=N_b}$为边界上的采样点,总采样点为 $N=N_f+N_b$。

本文算法跟 Ben-Shaul 等[21]的方法不同点在于:(1)构造方程残差损失函数时,在方程的两边除以了特征值的近似值 $R(\tilde{u})$;(2)损失函数的所有项都除以了特征函数平方的积分的近似值。如此构造的损失函数的各个部分之间的大小具有一致性,损失函数不会随着特征值的变动而剧烈变化,从而有利于选取权重 α、β、ρ 的值,也能提高训练的收敛性。

1.2.2 求解多个特征值问题

给定初始特征值 λ_0,求解最靠近此初始值的 m 个特征值和对应的特征函数。令 $\tilde{u}_i(X;\theta)\ (i=1,2,\cdots,m)$ 为神经网络的 m 个输出,分别近似一个特征函数。构造损失函数如下:

$$\begin{aligned} L(\theta)=&\sum_{i=1}^{m}L_{\text{eq}}^{(i)}(\theta)+\alpha\sum_{i=1}^{m}L_{\text{bd}}^{(i)}(\theta)+\beta\sum_{i=1}^{m}L_{\text{unit}}^{(i)}(\theta)\\ &+\gamma\sum_{k=i+1}^{m}\sum_{i=1}^{m}L_{\text{orth}}^{(i,k)}(\theta)+\rho\sum_{i=1}^{m}L_{\text{lambd}}^{(i)}(\theta), \end{aligned} \tag{6}$$

$$L_{\text{orth}}^{(i,k)}(\theta)=\frac{\left[\dfrac{1}{N_f}\displaystyle\sum_{j=1}^{N_f}\tilde{u}_i(X_j^f;\theta)\tilde{u}_k(X_j^f;\theta)\right]^2}{I(\tilde{u}_i)I(\tilde{u}_k)},$$

$L_{\text{eq}}^{(i)}(\theta)$、$L_{\text{bd}}^{(i)}(\theta)$、$L_{\text{unit}}^{(i)}(\theta)$、$L_{\text{lambd}}^{(i)}(\theta)$ 分别为第 i 个特征函数 $\tilde{u}_i$ 的方程残差损失函数、边界损失函数、归一化损失函数、特征值损失函数。计算方法跟上面类似,只需将 $\tilde{u}_i$ 代替 $\tilde{u}$。$L_{\text{orth}}^{(i,k)}(\theta)$ 为第 i 个特征函数 $\tilde{u}_i$ 和第 k 个特征函数 $\tilde{u}_k$ 的正交性损失函数。

相对于 Ben-Shaul 等[21]的方法,本文方法中的方程残差损失函数不会随特征值变化而剧烈变化,更有利于求解多个特征值问题。

2 数值实验

数值实验是在 Colab 上进行运算,神经网络架构均采用全连接方式,在不特别说明的情况下,神经网络隐藏层为 5 层,激活函数选取正弦函数 $\sin(\frac{\pi}{2}x)$,输入值归一到 $(-1,1)$,优化器选择为 Adam,使用指数衰减学习率,初始学习率取 0.001,每 100 次进行一次学习率的衰减,衰减系数为 0.9,最小学习率为 2×10^{-6},所有的计算结果都是重复计算 3 次的平均值。

2.1 一维特征值问题

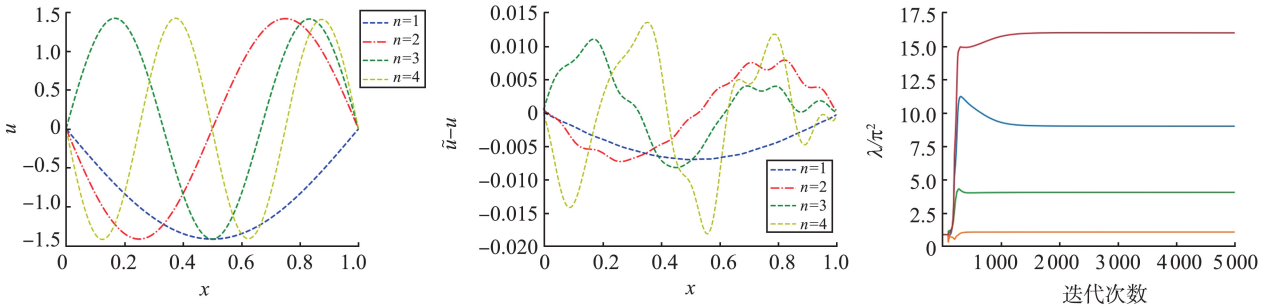
实验 1 一维若干个绝对值最小特征值

考虑微分方程

$$\begin{cases} u''(x)=-\lambda u(x), & x\in(0,1),\\ u(0)=u(1)=0 \end{cases} \tag{7}$$

的一维特征值问题,方程的特征值为 $\lambda_n=(n\pi)^2$,对应的特征函数为 $u_n(x)=\sqrt{2}\sin(n\pi x)$, $n=1,2,3,\cdots$。

以计算前 4 个绝对值最小特征值为例,取初始值 $\lambda_0=0.8\pi^2$,计算结果如图 1 所示。图 1(a)中每条曲线代表一个特征函数,这些函数的形状类似正弦波,反映出在区间内的振荡特性,且频率逐渐增大,与问题的解析特征函数一致;图 1(b)展示了上述特征函数与其真解之间的误差分布,误差曲线清晰地刻画了各特征函数在整个区间内的误差情况,可以看出,误差非常小,证明了本文方法在近似特征函数时的高精度;图 1(c)展示了本文方法计算得到的前 4 个绝对值最小特征值,给出的特征值与理论真值非常接近,进一步验证了该方法在求解一维特征值问题上的准确性。特征函数的相对均方误差平均值和特征值的相对误差平均值如表 1 所示。可以发现本文与文献[21]在计算特征函数时相差不大,但本文方法在计算特征值时效果更好、更精确。



(a) 特征函数　(b) 特征函数的误差　(c) 特征值

图 1　本文计算一维前 4 个绝对值最小特征值问题的结果

Fig.1　Results of calculating the first four absolute minimum eigenvalue problems in one dimension in this article

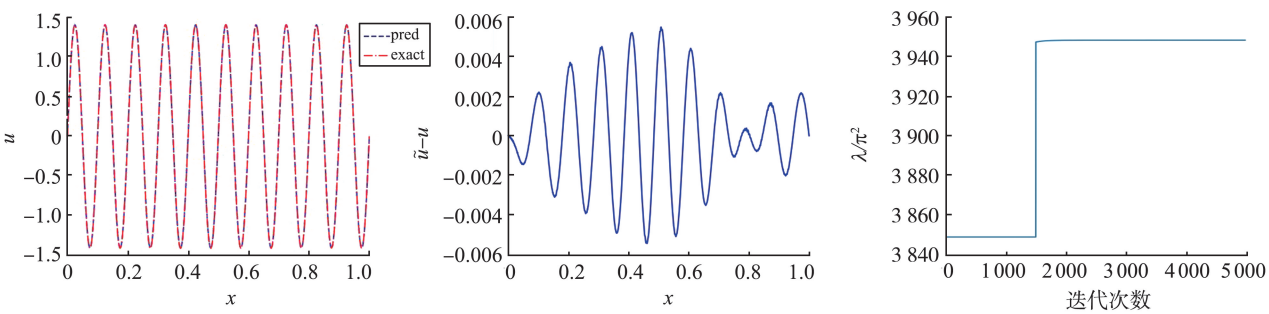
表 1　一维前 4 个绝对值最小特征值的数值结果与已有方法的比较

Table 1　Comparison of numerical results of the first four absolute minimum eigenvalues in one dimension with existing methods

方法	$\frac{\lambda}{\pi^2}$(真实值为 1,4,9,16)	特征值的相对误差平均值	特征函数的相对均方误差平均值
本文方法	(0.999 4,3.999 8,9.001 3,16.001 3)	1.41×10^{-4}	3.88×10^{-5}
文献[21]方法	(1.12,4.09,9.03,15.95)	4.00×10^{-2}	2.05×10^{-5}

实验 2　一维最靠近 λ_0 的特征值

当特征值很大时,对应的特征函数震荡非常厉害,训练过程通常很难收敛。针对此问题,本文利用两阶段训练法和多尺度神经网络。整个训练分为 2 个阶段。第一阶段:在前 K_0 次训练中,不用瑞利商(2)更新特征值,而是在损失函数式(5)、(6)中用初始值 λ_0 代替 $R(\tilde{u})$。这样做是因为在训练初期特征函数的误差很大,用瑞利商(2)计算出的特征值误差很大。第二阶段:从 K_0+1 次训练开始,每次迭代时用瑞利商(2)更新特征值。此时的特征函数已经有一定的精度,利用瑞利商(2)计算出的特征值也应该有一定的精度。以计算特征值 $\lambda_{20}=400\pi^2$ 为例,取 $\lambda_0=390\pi^2$。取采样点 $N_f=800$,隐含层神经元个数为 80,多尺度神经网络参数 $M=40$,第一阶段训练 $K_0=100$ 次。计算结果如图 2 所示,图 2(a)显示了利用本文方法计算出的特征函数与其对应的真解,尽管大特征值对应的特征函数振荡很强,但计算得到的特征函数与真解高度吻合,证明了采用两阶段训练法及多尺度神经网络后,在计算大特征值问题时也能够获得较为准确的函数近似;图 2(b)展示了特征函数近似值与真解之间的误差分布,图中误差曲线反映出在整个定义域内误差均保持在极小范围内,这说明在经过初步训练后,网络已学到足够的特征信息,从而保证了后续用瑞利商更新特征值时误差不会因初始的大误差而放大;图 2(c)呈现了采用两阶段训练法后计算得到的特征值,该图显示了特征值在第二阶段训练过程中逐步收敛,最终逼近真实特征值,这验证了在初期阶段先保持特征函数的稳定,再利用瑞利商更新特征值的策略,有效提高了求解大特征值问题的精度。



(a) 特征函数与真解　(b) 特征函数的误差　(c) 特征值

图 2　本文计算特征值 $\lambda_{20}=400\pi^2$ 和特征函数的结果

Fig.2　Results of eigenvalue $\lambda_{20}=400\pi^2$ and eigenfunction are calculated in this paper

为了研究采样点数量 N_f 与隐含层神经元数量对计算结果的影响,本文计算了采样点和隐含层神经元数量取不同值时的结果,如表 2 所示。表 2 中列出了所计算的特征函数均方误差和特征值的相对误差。例如采样点为 $N_f=400$,隐含层神经元个数为 40 时,计算出的特征函数均方误差为 7.38×10^{-2},特征值相对误差

7.04×10^{-3}。从表 2 中可以看出,误差随着采样点数量和隐含层神经元数量增加而减少。固定神经元数量,增加采样点数量,误差收敛非常快。表 3 显示了多尺度参数对计算结果的影响,取采样点 $N_f=800$,隐含层神经元个数为 80,可以看出多尺度参数的取值不是越大越好,当取值过大时,误差开始变大。

表 2 神经元个数和采样点数量对误差的影响

Table 2 Effect of number of neurons and number of sampling points on error

采样点数量 N_f	神经元个数 n	特征函数均方误差	特征值相对误差
200	40	1.640	6.06×10^{-1}
	60	1.597	3.89×10^{-2}
	80	1.010	2.68×10^{-2}
	100	1.422	1.47×10^{-2}
400	40	7.38×10^{-2}	7.04×10^{-3}
	60	7.38×10^{-4}	1.23×10^{-3}
	80	1.24×10^{-4}	3.50×10^{-4}
	100	1.70×10^{-5}	4.52×10^{-5}
800	40	2.94×10^{-1}	3.98×10^{-2}
	60	3.37×10^{-5}	3.15×10^{-6}
	80	2.85×10^{-6}	3.06×10^{-6}
	100	4.16×10^{-6}	1.74×10^{-6}

表 3 多尺度参数对误差的影响

Table 3 Effect of multiscale parameters on error

M	特征值相对误差	特征函数均方误差
1	1.597	0.273
10	3.51×10^{-5}	1.10×10^{-6}
20	1.84×10^{-6}	6.48×10^{-7}
40	1.42×10^{-5}	4.36×10^{-6}
60	6.673	0.255

2.2 二维正方形区域特征值问题

实验 3 二维前若干个绝对值最小特征值

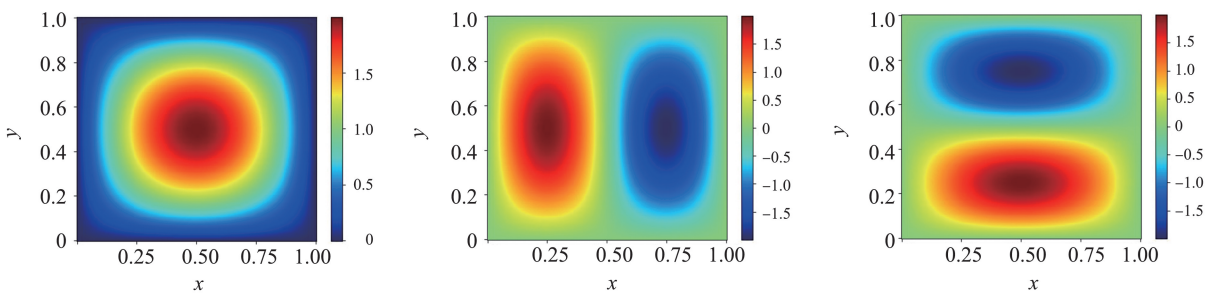
考虑方程

$$\begin{cases}\Delta u(x,y)=-\lambda u(x,y), & (x,y)\in\Omega,\\ u|_{\partial\Omega}=0\end{cases} \tag{8}$$

的二维特征值问题,这里 $\Omega=(0,1)^2$。方程的特征值为 $\lambda_{n,m}=(n^2+m^2)\pi^2$,对应的特征函数为

$$u_{n,m}(x,y)=2\sin(n\pi x)\sin(m\pi y)\ 。 \tag{9}$$

以计算前 4 个绝对值最小特征值为例,取初始值 $\lambda_0=1.8\pi^2$,计算结果如图 3 所示,图 3(a)—(d)显示了本文方法计算出的前 4 个特征函数,这些图直观地反映了所求二维特征函数的振荡模式和空间分布情况,验证了本文方法在逼近解析解时的有效性;图 3(e)—(h)显示了本文方法计算出的前 4 个特征函数与真解的误差,从这些图中可以看出,各特征函数的数值解与真解之间的误差都非常小,说明本文方法在二维正方形区域内求解特征值问题时具有很高的精度;特征函数的相对均方误差平均值和特征值的相对误差的平均值如表 4 所示,可以发现本文方法比文献[21]方法在计算二维特征值问题时误差更小,效果更好。



(a) n=m=1的特征函数 (b) n=1, m=2的特征函数 (c) n=2, m=1的特征函数

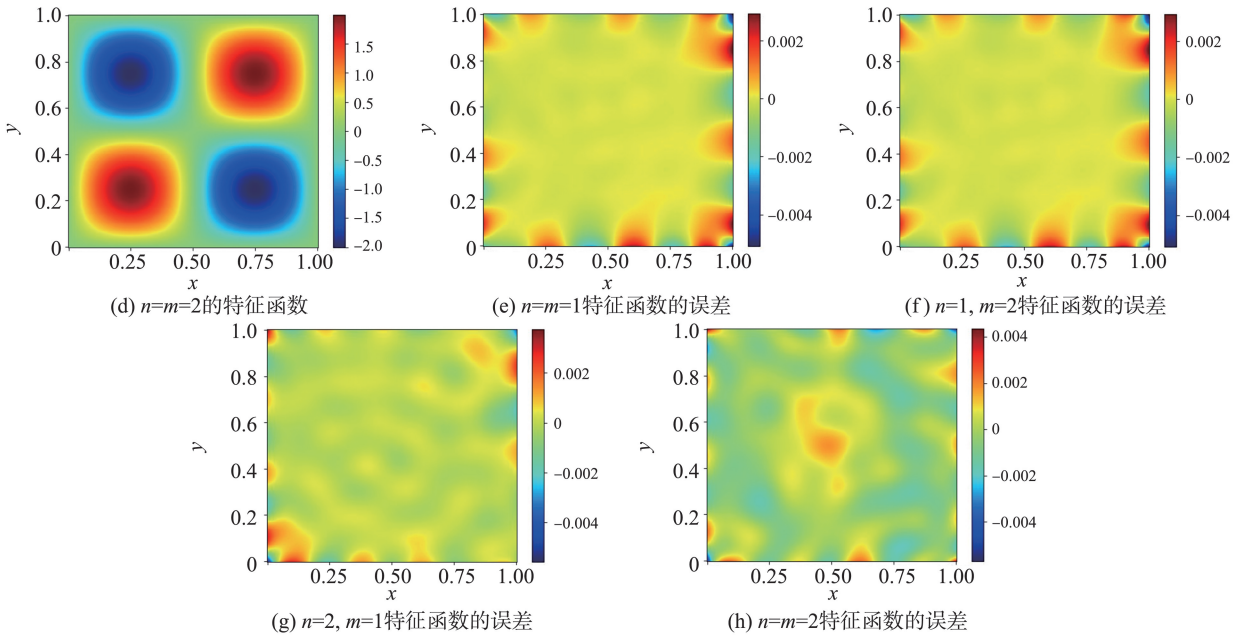


图 3　二维正方形区域中前 4 个绝对值最小特征值问题的结果

Fig.3　Results of the problem of the first four absolute minimum eigenvalues in a two-dimensional square region

表 4　二维正方形区域中前 4 个绝对值最小特征值的数值结果与已有方法的比较

Table 4　Comparison of numerical results of the first four absolute minimum eigenvalues in a two-dimensional square region with existing methods

方法	$\frac{\lambda}{\pi^2}$(真实值为 2,5,5,8)	特征值相对误差平均值	特征函数相对均方误差平均值
本文方法	(2.000 071, 4.999 628, 4.999 737, 7.999 533)	5.42×10^{-5}	2.83×10^{-7}
文献[21]方法	(1.98, 4.93, 4.96, 7.91)	1.00×10^{-2}	1.00×10^{-3}

实验 4　二维最靠近初始值 λ_0 的特征值

以计算特征值 $\lambda_{3,3}=18\pi^2$ 为例,取 $\lambda_0=18.5\pi^2$。多尺度神经网络参数 $M=5$,第一阶段训练 $K_0=1\,000$ 次。

计算结果如图 4 所示,图 4(a)中展示了利用本文方法计算得到的特征函数,这些数值解的特征函数在二维区域内的振荡和分布与真解非常吻合;图 4(b)显示了本文方法计算出的特征函数与真解的误差,从图中可以看出,整个区域内的误差均非常小,表明本文方法在求解特征函数时精度较高;图 4(c)展示了训练过程中损失函数的变化情况,可以看到,损失函数随着训练的进行逐步降低并收敛,这反映出网络训练的稳定性和有效性;图 4(d)展示了计算过程中更新得到的特征值,图中显示的特征值经过多次迭代后逐渐收敛到接近真实值的数值,证明了本文方法在求解特征值问题上的准确性。其求解的特征函数的均方误差为1.18×10^{-6},特征值的相对误差为 8.47×10^{-4}。

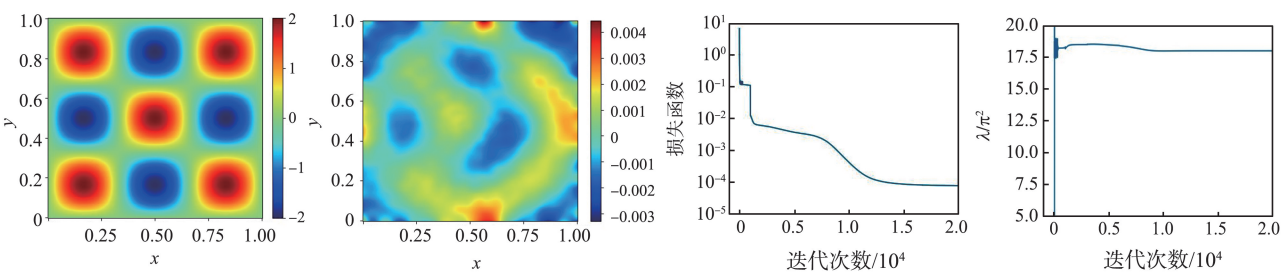


图 4　特征值 $\lambda_{3,3}=18\pi^2$ 和特征函数的结果

Fig.4　Results of eigenvalue $\lambda_{3,3}=18\pi^2$ and eigenfunction

实验 5　二维二重特征值

以计算特征值 $\lambda_{4,5}=\lambda_{5,4}=41\pi^2$ 为例,取 $\lambda_0=40\pi^2$。多尺度神经网络参数 $M=10$,第一阶段训练 $K_0=$

3 000 次。计算结果如图 5 所示,图 5(a)、5(b)分别给出了针对同一个二重特征值问题求解得到的两个正交特征函数,图 5(a)和图 5(b)展示的特征函数虽然对应同一特征值,但在空间分布和振荡模式上有所区别,反映了本文方法在求解重特征值问题方面的能力;图 5(c)、5(d)分别展示了 5(a)、5(b)中所求特征函数与其参考解之间的误差分布,误差图表明,两组特征函数在整个计算区域内的近似误差小,证明了本文方法在求解二重特征值问题时的有效性;图 5(e)展示了训练过程中损失函数的变化曲线,曲线显示出在训练初期损失较高,随后逐步下降并趋于稳定,说明网络在两阶段训练过程中逐渐收敛,确保了特征函数和特征值的精确计算;图 5(f)展示了利用本文方法计算得到的特征值的演变过程,随着训练的进行,特征值逐渐收敛到一个稳定值,并且该值与参考值非常接近,从而验证了本文方法在求解二维二重特征值问题时的精度和稳定性。特征函数的均方误差和特征值的相对误差如表 5 所示,从表 5 中可以发现,本文方法计算二维二重特征值问题时误差很小。

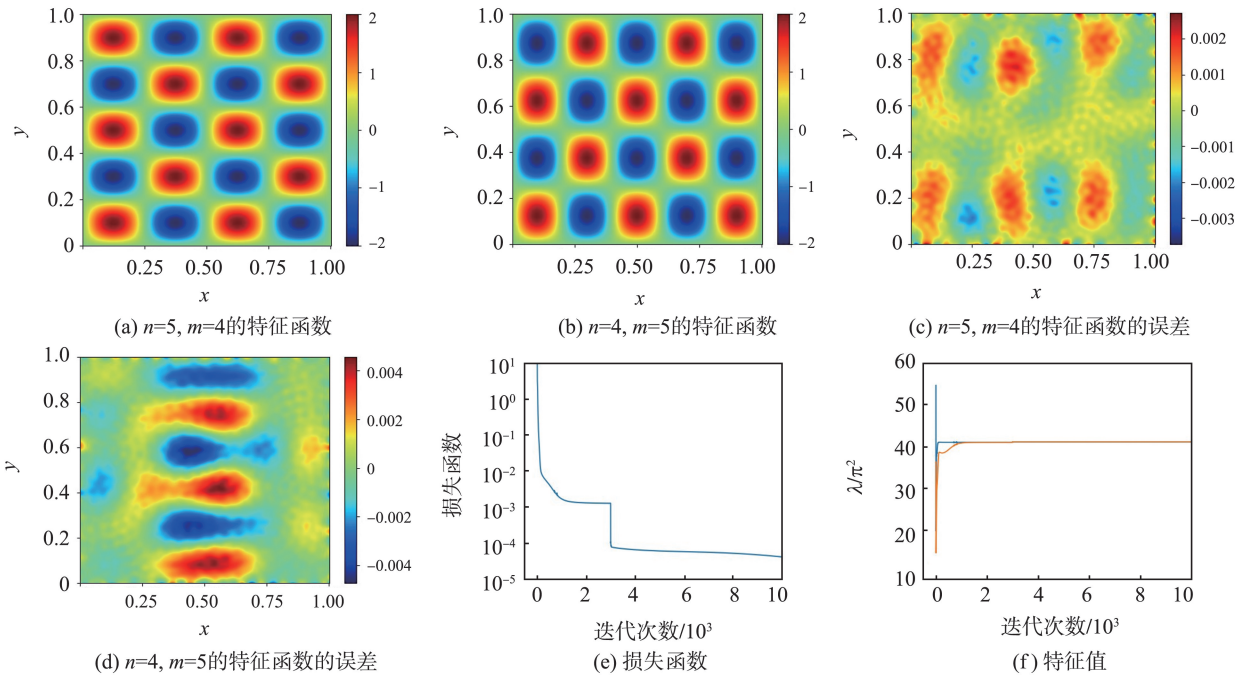


图 5 特征值 $\lambda_{4,5}=\lambda_{5,4}=41\pi^2$ 和特征函数的结果

Fig.5 Results of eigenvalue $\lambda_{4,5}=\lambda_{5,4}=41\pi^2$ and eigenfunction

表 5 求解二维二重特征值问题解的误差

Table 5 Error of solving the two-dimensional dual eigenvalue problem

误差	$n=5,\ m=4$	$n=4,\ m=5$
特征函数均方误差	5.46×10^{-7}	2.36×10^{-6}
特征值相对误差	6.49×10^{-6}	3.39×10^{-5}

2.3 二维 L 形区域特征值问题

实验 6 L 形区域前若干个绝对值最小特征值

在 L 形区域特征值问题中,方程(8)的区域变为 $\Omega=(0,1)^2\setminus(0,0.5)^2$。此特征值问题无解析解,用有限元方法计算出的数值结果作为参考值,以计算前 4 个绝对值最小特征值为例,取初始值 $\lambda_0=10$,计算结果如图 6 所示,图 6(a)—(d)这 4 幅图分别展示了利用本文方法计算得到的前 4 个绝对值最小特征值对应的特征函数,图中可以直观看到,在 L 形区域内各特征函数的空间分布和振荡模式,验证了本文方法对非规则区域(L 形)的适应性与高精度逼近能力;图 6(e)—(h)这 4 幅图展示了前述特征函数与参考解的误差分布情况,图中显示在整个 L 形区域内,各特征函数的近似误差均保持在较小范围内,证明了本文方法在求解该类问题时能够获得高精度的特征函数近似。特征函数的均方误差和特征值的相对误差如表 6 所示,可以发现本文方法比文献方法[21]在计算二维 L 形特征值问题时误差更小,效果更好。

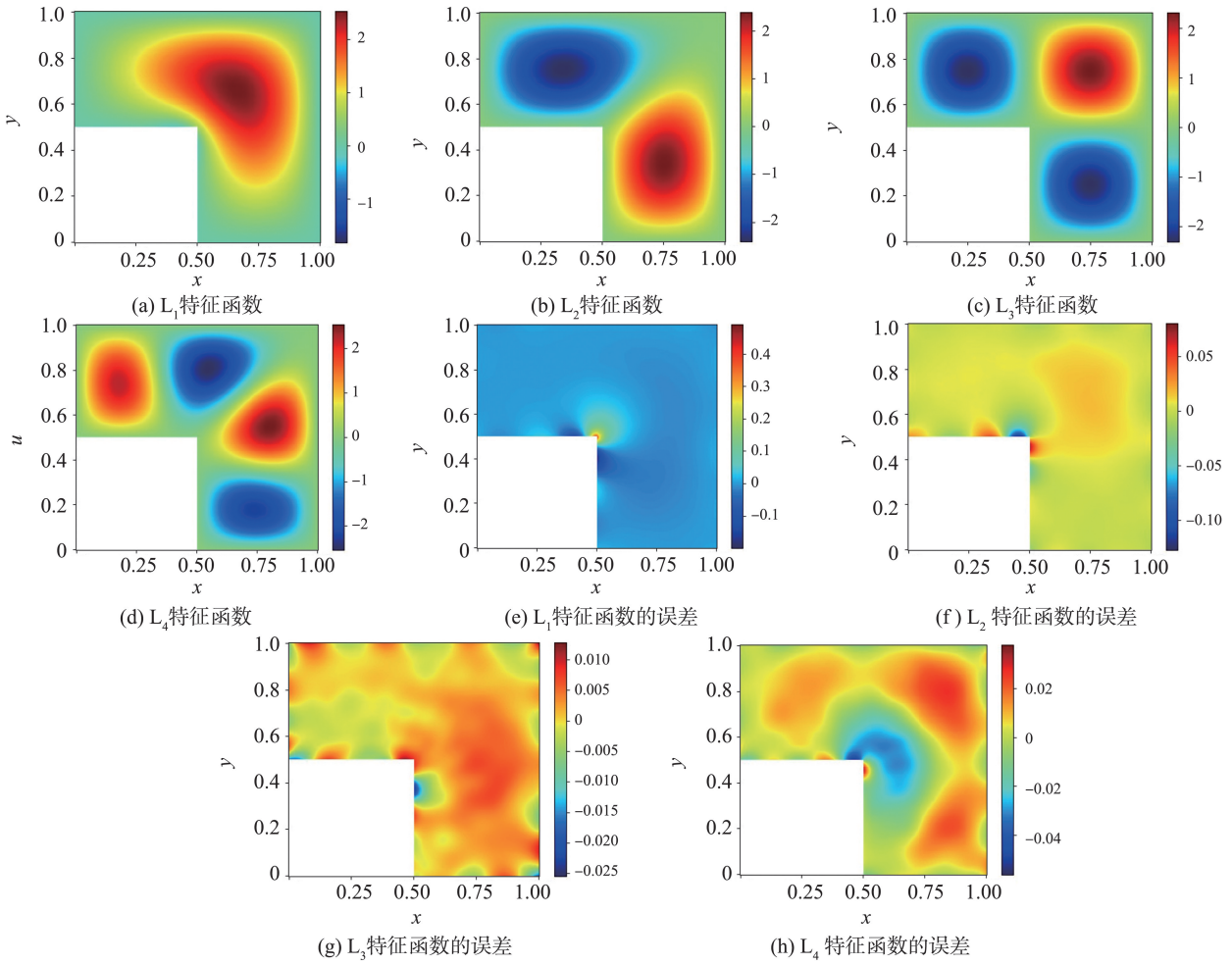


图 6　L 形区域中前 4 个绝对值最小特征值问题的结果

Fig.6　Results of the problem of the first four absolute minimum eigenvalues in an L-shaped region

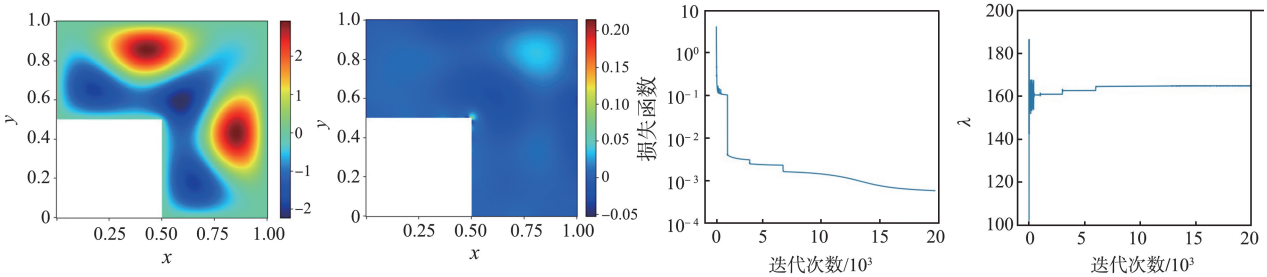
表 6　L 形区域中前 4 个绝对值最小特征值的数值结果与已有方法的比较

Table 6　Comparison of numerical results of the first four absolute minimum eigenvalues in the L-shaped region with existing methods

方法	$\frac{\lambda}{4}$	特征值相对误差	特征函数均方误差
本文方法	(9.506, 15.170, 19.743, 29.503)	$(1.39\times10^{-2}, 8.76\times10^{-4}, 3.41\times10^{-4}, 3.78\times10^{-4})$	$(3.99\times10^{-4}, 7.20\times10^{-5}, 7.42\times10^{-6}, 1.24\times10^{-4})$
文献[21]方法	(8.626, 14.711, 19.288, 28.901)	$(1.06\times10^{-1}, 3.23\times10^{-2}, 2.32\times10^{-2}, 2.16\times10^{-2})$	$(2.0\times10^{-3}, 2.0\times10^{-3}, 3.0\times10^{-3}, 3.0\times10^{-3})$

实验 7　L 形区域最靠近 λ_0 的特征值

以计算特征值 $\lambda_6=165.912$ 为例,取 $\lambda_0=160$。多尺度神经网络参数 $M=5$,第一阶段训练 $K_0=1\,000$ 次。计算结果如图 7 所示,图 7(a)显示了本文方法计算得到的特征函数在 L 形区域内的空间分布,图中所示的数值解反映了特征函数在不规则区域内的振荡和分布特性,证明了本文方法对不规则几何区域的适应能力;图 7(b)展示了计算得到的特征函数与参考解之间的误差分布,从图中可以看出,整个区域内误差均较小,说明所求特征函数具有较高精度,误差控制在较低水平;图 7(c)记录了训练过程中损失函数的变化情况。随着训练的进行,损失值逐步下降并最终趋于稳定,这表明网络训练过程收敛顺利,有效地逼近了真实特征函数;图 7(d)显示了特征值在训练过程中不断更新并逐步收敛到最终值的情况,最终计算得到的特征值与参考解相当接近,验证了本文方法在求解该 L 形区域最靠近 λ_0 的特征值问题上的准确性。其求解的特征函数的均方误差为 9.22×10^{-5},特征值的相对误差为 7.80×10^{-3}。



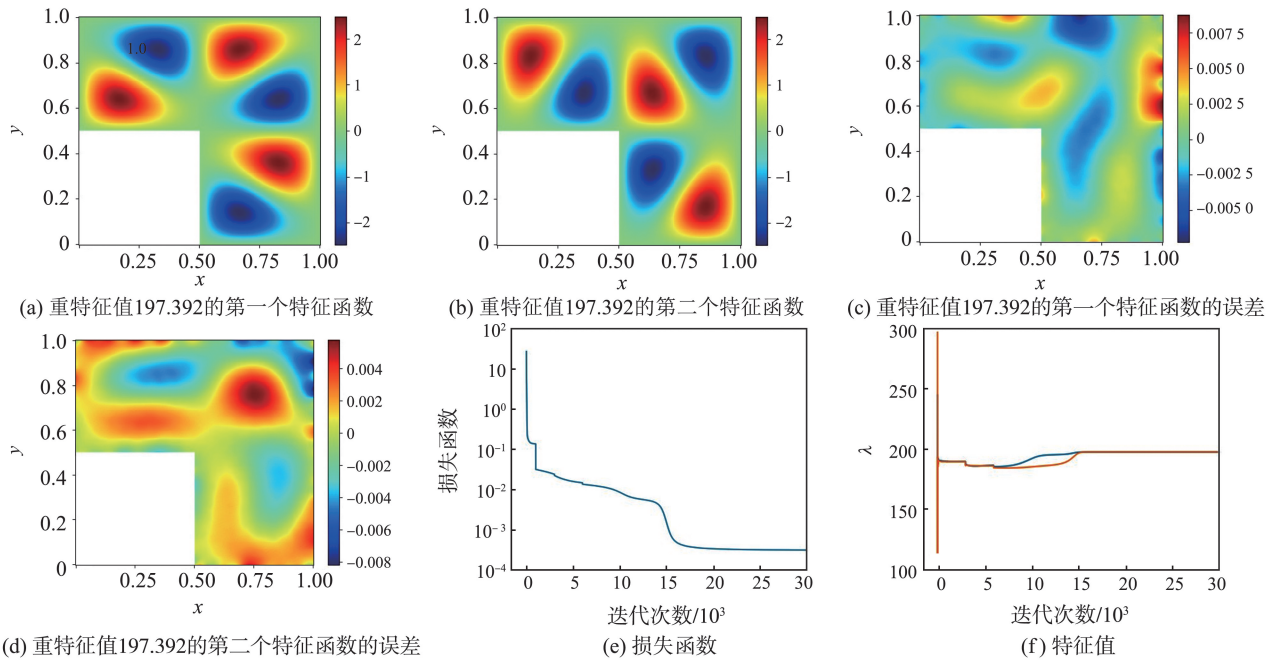
(a) 特征函数 (b) 特征函数的误差 (c) 损失函数 (d) 特征值

图 7 L 形区域中特征值 $\lambda_6 = 165.912$ 和特征函数的结果

Fig.7 Results of the eigenvalue $\lambda_6 = 165.912$ and the eigenfunction in the L-shaped region

实验 8 L 形区域二重特征值

以计算重特征值 197.392 为例,取 $\lambda_0 = 190$。多尺度神经网络参数 $M = 8$,第一阶段训练 $K_0 = 1\,000$ 次。计算结果如图 8 所示,图 8(a)、(b)这两幅图分别展示了针对同一重特征值问题求解得到的两个正交特征函数,图 8(a)和图 8(a)反映出不同的振荡模式和空间分布,即便它们对应相同的特征值,但其结构特性仍有所不同;图 8(c)、(d)这两幅图展示了图 8(a)和图 8(b)中各特征函数与参考解之间的误差分布情况,可以看出,误差在整个 L 形区域内均保持在较低水平,证明了本文方法在逼近 L 形区域二重特征值问题中的特征函数时具有很高的精度;图 8(e)记录了训练过程中损失函数的变化情况,曲线显示出在训练初期损失较高,随后逐渐下降并趋于稳定,表明网络训练过程顺利收敛,为特征函数和特征值的精确求解奠定了基础;图 8(f)展示了训练过程中计算得到的特征值演变情况,特征值逐步更新并最终收敛到一个稳定值,与参考解非常接近,从而验证了本文方法在求解 L 形区域二重特征值问题上的准确性。特征函数的均方误差和特征值的相对误差如表 7 所示,从表中可以看出我们方法对计算二维 L 形区域特征值问题误差很小。



(a) 重特征值197.392的第一个特征函数 (b) 重特征值197.392的第二个特征函数 (c) 重特征值197.392的第一个特征函数的误差

(d) 重特征值197.392的第二个特征函数的误差 (e) 损失函数 (f) 特征值

图 8 L 形区域中重特征值 197.392 和特征函数的结果

Fig.8 Results of calculating the heavy eigenvalue 197.392 and the eigenfunction in the L-shaped region

表 7 L 形区域二重特征值问题解的误差

Table 7 Error in solving the dual eigenvalue problem in L-shaped regions

误差	重特征值 1	重特征值 2
特征函数的均方误差	3.39×10^{-6}	3.66×10^{-6}
特征值的相对误差	1.71×10^{-3}	1.79×10^{-3}

从上述几个实验可以看出本文方法不但比现有方法求解绝对值最小的特征值的精度更高,而且还能求解现有方法不能求解的大特征值问题和重特征值问题。

实验 9　计算效率比较

以计算式(8)最小特征值为例,采用文献[22]的参数,即神经网络层数为 7 层、每个隐含层的神经元个数为 100 个、区域内部随机采样点为 5 000 个、边界随机采样点为 500 个、学习率为 0.003 等情况下,两种方法运算的时间和误差如表 8 所示,从表中可以看出,由于本文方法构造的损失函数比现有方法简单,因此计算效率更快、误差更小。

表 8　本文方法与文献方法的误差和运算时间的比较

Table 8　Comparison of calculation time and error between the method in this article and the method in literature

方法	特征函数的均方误差	特征值的相对误差	运算时间/s
文献[21]方法	1.66×10^{-2}	6.27×10^{-2}	123.18
本文方法	2.40×10^{-3}	3.08×10^{-3}	117.94

本文数值实验中神经网络的超参数见表 9 和表 10。

表 9　数值实验超参数

Table 9　Numerical experiment hyperparameter

实验	每层神经元个数	内部均匀采样点数量 N_f	边界随机采样点数量 N_b	训练次数 K	多尺度参数 M
实验 1	60	200		5 000	4
实验 2	80	800		5 000	40
实验 3	100	10 000	400	20 000	3
实验 4	100	10 000	400	20 000	5
实验 5	100	10 000	400	10 000	10
实验 6	100	30 000	600	20 000	3
实验 7	100	30 000	600	20 000	5
实验 8	100	30 000	600	30 000	8

表 10　损失函数权重取值

Table 10　Loss function weight value

实验	α	β	γ	ρ
实验 1	1	1	10	0
实验 2	1	1	0	0
实验 3	1	1	1	0
实验 4	1	1	0	10
实验 5	1	1	1	0
实验 6	1	1	1	0
实验 7	1	1	1	10
实验 8	1	1	1	10

3　总结

本文通过对损失函数进行无量纲化处理和利用两阶段训练法提出一个改进物理信息神经网络法来求解微分方程特征值问题。此方法可以求解多个绝对值最小特征值、最靠近给定初始的单个和多个特征值,能求解大特征值问题,相比于现有算法更有效,精度更高。由于本文需要用到正交性,只能用于求解自伴微分算子的多个特征值问题。下一步需要推广此算法来求解没有正交性的非自伴算子的特征值问题。

参考文献:

[1] 李蕾,叶永升. 具有 Dirichlet 有界条件的反应扩散 Cohen-Grossberg 神经网络指数稳定性[J]. 山东大学学报(理学版),2023,58(10):67-74.

LI Lei, YE Yongsheng. Exponential stability of reaction-diffusion Cohen-Grossberg neural networks with Dirichlet boundary conditions[J]. Journal of Shandong University (Natural Science), 2023, 58(10):67-74.

[2] 王新生,朱小飞,李程鸿. 标签指导的多尺度图神经网络蛋白质作用关系预测方法[J]. 山东大学学报(理学版),2023,58(12):22-30.
WANG Xinsheng, ZHU Xiaofei, LI Chenghong. Label guided multi-scale graph neural network for protein-protein interaction prediction[J]. Journal of Shandong University (Natural Science), 2023, 58(12):22-30.

[3] 徐光柱,刘鸣,任东,等. 基于脉冲耦合神经网络的多区域图像分割[J]. 山东大学学报(理学版),2010, 45(7):86-93.
XU Guangzhu, LIU Ming, REN Dong, et al. Multi-region image segmentation based on a pulse coupled neural network[J]. Journal of Shandong University (Natural Science), 2010, 45(7):86-93.

[4] HUANG S D, FENG W T, TANG C W, et al. Partial differential equations meet deep neural networks: a survey [EB/OL]. (2022-03-17)[2025-03-18]. https://arxiv.org/abs/2211.05567v2.

[5] E W N . The dawning of a new era in applied mathematics[J]. Notices of the American Mathematical Society, 2021, 68(4): 565-571

[6] HAN J Q, JENTZEN A, E W N. Solving high-dimensional partial differential equations using deep learning[J]. Proceedings of the National Academy of Science, 2018, 115(34):8505-8510.

[7] E W N, YU B. the deep Ritz method: a deep learning-based numerical algorithm for solving variational problems[J]. Communications in Mathematics and Statistics, 2018, 6(1):1-12.

[8] SIRIGNANO J, SPILIOPOULOS K. DGM: a deep learning algorithm for solving partial differential equations[J]. Journal of Computational Physics, 2018, 375:1339-1364.

[9] 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.

[10] ZANG Y H, BAO G, YE X J, et al. Weak adversarial networks for high-dimensional partial differential equations[J]. Journal of Computational Physics, 2020, 411:109409.

[11] HAO Z K, LIU S M, ZHANG Y C, et al. Physics-informed machine learning: a survey on problems, methods and applications[EB/OL].(2023-03-07)[2025-03-18]. https://arxiv.org/abs/2211.08064.

[12] CAI S Z, MAO Z P, WANG Z C, et al. Physics-informed neural networks (PINNs) for fluid mechanics: a review[J]. Acta Mechanica Sinica, 2021, 37(12):1727-1738.

[13] CUOMO S, DICOLA V S, GIAMPAOLO F, et al. Scientific machine learning through physics-informed neural networks: where we are and what's next[J]. Journal of Scientific Computing, 2022, 92(3):88.

[14] RATHORE P, LEI W M, FRANGELLA Z, et al. Challenges in training PINNs: a loss landscape perspective[EB/OL]. (2024-06-03)[2025-03-18]. https://arxiv.org/abs/2402.01868.

[15] ANAGNOSTOPOULOS S J, TOSCANO J D, STERGIOPULOS N, et al. Learning in PINNs: phase transition, total diffusion, and generalization[EB/OL]. (2024-03-27)[2025-03-18]. https://arxiv.org/abs/2403.18494.

[16] LU L, MENG X H, MAO Z P, et al. DeepXDE: a deep learning library for solving differential equations[J]. SIAM Review, 2021, 63(1):208-228.

[17] LU L, JIN P Z, PANG G F, et al. Learning nonlinear operators via Deep ONet based on the universal approximation theorem of operators[J]. Nature Machine Intelligence, 2021, 3:218-229.

[18] CAO Q Y, GOSWAMI S, KARNIADAKIS G E. Laplace neural operator for solving differential equations[J]. Nature Machine Intelligence, 2024, 6:631-640.

[19] HAN J Q, LU J F, ZHOU M. Solving high-dimensional eigenvalue problems using deep neural networks: a diffusion Monte Carlo like approach[J]. Journal of Computational Physics, 2020, 423:109792.

[20] YANG Q H, DENG Y T, YANG Y, et al. Neural networks based on power method and inverse power method for solving linear eigenvalue problems[J]. Computers & Mathematics with Applications, 2023, 147:14-24.

[21] BEN-SHAUL I, BAR L, FISHELOV D, et al. Deep learning solution of the eigenvalue problem for differential operators[J]. Neural Computation, 2023, 35(6):1100-1134.

[22] LIU Z Q, CAI W, XU Z J. Multi-scale deep neural network (MscaleDNN) for solving Poisson-Boltzmann equation in complex domains[J]. Communications in Computational Physics, 2020, 28(5):1970-2001.

(编辑:胡春燕)