

基于残差的 Fourier 神经网络求解 广域椭圆型微分方程以及收敛性证明^{*}

颜景越, 苏 欢

(哈尔滨工业大学(威海)理学院, 山东 威海 264209)

摘要: 物理信息神经网络(physics-informed neural network, PINN)在求解偏微分方程的正反问题中应用广泛.然而,传统 PINN 在求解广域微分方程时,存在精度不足、易陷入局部最优解的局限,且缺乏收敛性理论保证.针对上述问题,提出了一种基于残差的 Fourier 神经网络.该网络将残差 Fourier 层集成到传统 PINN 框架中,利用三角函数的周期性特征有效提升模型在求解广域问题时的精度.特别地,针对线性椭圆型微分方程,建立了基于残差的 Fourier 神经网络的收敛性理论分析.数值实验结果表明,基于残差的 Fourier 神经网络在求解正反问题时,相较于传统 PINN,具有更高的求解精度和更快的收敛速度.

关键词: 物理信息神经网络; 残差; Fourier 神经网络; 广域微分方程; 线性椭圆型微分方程; 收敛性分析

中图分类号: O241; TP183

文献标志码: A

DOI: 10.21656/1000-0887.460125

Residual-Based Fourier Neural Networks for Solving Elliptic Differential Equations on Unbounded Domains and Convergence Analysis

Yan Jingyue, Su Huan

(School of Science, Harbin Institute of Technology(Weihai),
Weihai, Shandong 264209, P.R. China)

Abstract: Physics-informed neural network (PINN) are widely applied for solving both forward and inverse problems of differential equations. However, traditional PINNs exhibit limitations when solving differential equations over large domains, including insufficient accuracy, susceptibility to local optima, and a lack of theoretical guarantees of convergence. To address these issues, a residual-based Fourier neural network (Res-FNN) was proposed. This network integrates residual Fourier layers into the traditional PINN framework, leveraging the periodicity property of trigonometric functions to enhance the model's accuracy in solving large-domain problems effectively. In particular, a theoretical convergence analysis was established for the Res-FNN applied to linear elliptic differential equations. Numerical experiments demonstrate that, the Res-FNN outperforms the traditional PINN in solving both forward and inverse problems, achieving higher accuracy and faster conver-

* 收稿日期: 2025-06-08; 修订日期: 2025-09-02

基金项目: 山东省自然科学基金面上项目(ZR202102220411)

作者简介: 颜景越(2001—),男,硕士生(通信作者. E-mail: 23s030162@stu.hit.edu.cn);

苏欢(1981—),女,副教授,博士,博士生导师(E-mail: suhuan@hitwh.edu.cn).

引用格式: 颜景越, 苏欢. 基于残差的 Fourier 神经网络求解广域椭圆型微分方程以及收敛性证明[J]. 应用数学和力学, 2026, 47(7): 912-923.

gence speed.

Key words: physics-informed neural network; residual; Fourier neural network; large-domain differential equation; linear elliptic differential equation; convergence analysis

0 引言

物理信息神经网络(PINN)^[1]在求解偏微分方程正反问题中受到人们的关注.PINN 将先验的物理知识融合到神经网络训练过程中,通过最小化残差损失函数来求解偏微分方程的正反问题.PINN 已经被广泛应用到空气动力学^[2-3]和流体力学^[4-5]等领域.例如:Mao 等^[3]探究了基于 PINN 来近似求解高速气动流动 Euler 方程的可行性;Jin 等^[4]提出了 NSFnet,把 PINN 应用到求解不可压缩 N-S 方程.与此同时,有关 PINN 求解偏微分方程的相关理论也在不断完善,如 Shin 等^[6]证明了 PINN 求解线性椭圆型偏微分方程和线性抛物型偏微分方程的收敛性.然而 PINN 仍然具有一些局限性,如 PINN 求解广域微分方程问题时往往伴随着大量的训练点,训练点的增多就会导致边界条件对训练神经网络约束的减小,使得训练结果非常不稳定,并且 PINN 对于具有高度振荡性或复杂时空特征的偏微分方程求解效果较差.

近年来,研究者已提出了多种方法克服 PINN 在广域问题中的局限性.例如:Kharazmi 等^[7]提出的 hp-variational physics-informed neural networks(hp-VPINNs),结合了变分法与区域分解的方法求解长区域微分方程;Jagtap 等^[8]提出了灵活分解域的 extended physics-informed neural networks(XPINNs)方法,划分的区域更加灵活;闵建等^[9]将课程-迁移学习与 PINN 相结合,用于求解曲面长时间扩散行为.神经网络相当于一个函数逼近器,不同的神经网络结构对应了不同的逼近函数,比如传统 PINN 使用 tanh 作为激活函数,它的函数逼近能力可以用通用近似定理^[10]来证明.与其类似的:Liu 等^[11]基于 K-A 表示定理^[12]提出了 KANs 神经网络;Liu^[13]提出了基于 Fourier 变换的 Fourier 神经网络.由于 Fourier 变换对周期函数、高频函数具有强大的表达能力,因此在处理周期边界的高频或广域问题时,可以采用 Fourier 神经网络提高模型的训练效果.Ngom 等^[14]提出了一个模糊的 Fourier 神经网络,他们基于一元 Fourier 变换构造了一个单隐层的神经网络.但是,由于输入 x 处在 $(2k+1)\pi$ 处时正弦函数的梯度为 0,神经网络隐藏层间很容易丢失重要信息,同时容易出现梯度消失的情况.

本文为了解决 PINN 在广域问题上容易出现的精度低的问题,同时提高 Fourier 神经网络的训练速度,提出了基于残差的 Fourier 神经网络.首先构造了基于残差的 Fourier 层,将基于残差的 Fourier 层集成到传统 PINN 结构上.然后对 Fourier 神经网络求解线性椭圆型偏微分方程的收敛性进行分析.最后通过数值实验验证了基于残差的 Fourier 神经网络求解偏微分方程正问题的优势和求解反问题的鲁棒性.

1 物理信息神经网络

给定以下形式的微分方程:

$$\begin{cases} \mathcal{A}[u(\mathbf{x}); \lambda] = f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \mathcal{B}[u(\mathbf{x})] = g(\mathbf{x}), & \mathbf{x} \in \partial\Omega, \end{cases} \quad (1)$$

其中, $\mathcal{A}, \mathcal{B}$ 是微分算子, $\mathbf{x} \in \mathbb{R}^m$. PINN 使用 L 层的全连接神经网络去近似微分方程的解 $u(\mathbf{x})$. 假设神经网络的输出为 $N(\mathbf{x}; \boldsymbol{\theta})$, 其中 N 代表神经网络, $\boldsymbol{\theta}$ 是神经网络中可训练的参数. $N(\mathbf{x}; \boldsymbol{\theta})$ 由下面公式给出:

$$N(\mathbf{x}; \boldsymbol{\theta}) = \sigma(\mathbf{W}^{(L)} \sigma(\mathbf{W}^{(L-1)} \cdots \sigma(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}) \cdots + \mathbf{b}^{(L-1)}) + \mathbf{b}^{(L)}), \quad (2)$$

其中, $\mathbf{W}^{(l)}$ 和 $\mathbf{b}^{(l)}$ 是第 l 层的权重矩阵和偏置, $\boldsymbol{\theta} = (\mathbf{W}^{(l)}, \mathbf{b}^{(l)}), l = 1, 2, \dots, L$; σ 是激活函数,常用的激活函数有 sigmoid 函数、tanh 函数等.

损失函数共分为三部分,即数据损失 E_d 、方程损失 E_f 和边界损失 E_b , E 表示均方差(MSE):

$$\begin{cases} E = E_d + E_f + E_b, \\ E_d = \frac{1}{N_d} \sum_{i=1}^{N_d} |N(\mathbf{x}_{di}; \boldsymbol{\theta}) - u_i(\mathbf{x}_{di})|^2, \\ E_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |N(\mathbf{x}_{fi}; \boldsymbol{\theta}) - \mathcal{N}(N; \lambda)|^2, \\ E_b = \frac{1}{N_b} \sum_{i=1}^{N_b} |N(\mathbf{x}_{bi}; \boldsymbol{\theta}) - g(\mathbf{x}_{bi})|^2, \end{cases} \quad (3)$$

其中, N_d 是观测点的数量, N_f 是区域 Ω 中样本点的数量, N_b 是边界处样本点的数量, $\mathbf{x}_{di}$ 是观测点, $\mathbf{x}_{fi}$ 是区域 Ω 中的样本点, $\mathbf{x}_{bi}$ 是边界处的样本点. 但是 PINN 处理广域或高频问题时会出现精度过低, 训练速度慢的问题. 下面给出一个例子:

$$\begin{cases} \frac{d^2 y}{dx^2} = -4\pi^2 y, \\ y(0) = 0, \quad \frac{dy}{dx}(0) = 2\pi, \end{cases} \quad (4)$$

此微分方程有解析解 $y = \sin(2\pi x)$.

取 x 的区间为 $[0, b]$, $b = 1, 1.5, \dots, 6$, 使用 PINN 进行训练, 然后比较每个区间上得到的误差, 为方便比较, 固定优化算法为 Adam 算法, 训练次数为 10 000 次.

图 1 展示了随着区间长度的增加, PINN 的误差不断积累导致精确度逐渐降低. 区间长度超过 3 时精确度减小的幅度很大, PINN 无法逼近此常微分方程的精确解. 图 2 展示了区间长度为 6 时, PINN 训练得到的结果与解析解的对比.

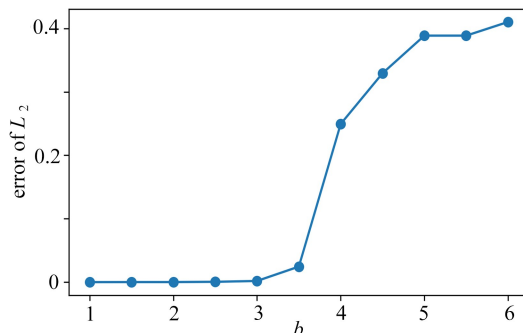


图 1 PINN 训练误差随区间长度 b 变化曲线

Fig. 1 The PINN training error variation with interval length b

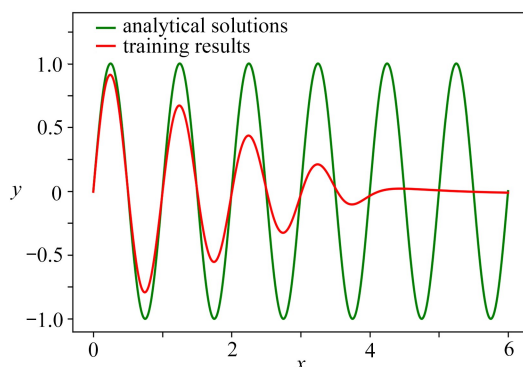


图 2 区间 $[0, 6]$ 中的 PINN 训练结果与解析解比较

Fig. 2 Comparison of PINN training results with analytical solutions in the interval $[0, 6]$

注 为了解释图中的颜色, 读者可以参考本文的电子网页版本, 后同.

从图 2 可以看到 PINN 训练结果与解析解的误差很大, 并且随着区间的变大, 训练结果的峰值不断减

小,在区间 $[4,6]$ 上,训练结果几乎趋近于一条直线,说明 PINN 更倾向于给出低频解^[15]。

为了解决 PINN 在长区域上求解此类微分方程时出现的精度低的问题,下面给出了适用于此类问题的神经网络结构——基于残差的 Fourier 神经网络。

2 基于残差的 Fourier 神经网络

提出了一个基于残差的 Fourier 神经网络(residual-based Fourier neural network, Res-FNN),首先定义多元实值函数的 Fourier 层:

Fourier 层表示为一个输入为 m 、输出为 n 的单隐层神经网络,隐藏层是由正弦函数和余弦函数组成的 Fourier 基函数表示,Fourier 层的表达式为

$$\mathbf{h}^{l+1}(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^m (w_{i,j}^{l+1} \cos(\mathbf{W}^l \mathbf{x}) + v_{i,j}^{l+1} \sin(\mathbf{V}^l \mathbf{x})) + \mathbf{b}^{l+1}, \quad (5)$$

其中, $\mathbf{W}^l \in \mathbb{R}^{m \times n}$, $\mathbf{V}^l \in \mathbb{R}^{m \times n}$ 和 $\mathbf{W}^{l+1} \in \mathbb{R}^{m \times n}$, $\mathbf{V}^{l+1} \in \mathbb{R}^{m \times n}$ 分别是第 l 层和第 $l+1$ 层的权重矩阵, $w_{i,j}^{l+1}$ 和 $v_{i,j}^{l+1}$ 分别是 $\mathbf{W}^{l+1}$ 和 $\mathbf{V}^{l+1}$ 的 (i,j) 元素,第 l 层的隐藏单元数量是 $2m$,可以看作是两个神经网络。

这样 Fourier 层的输出就是近似 Fourier 变换的形式。可以在原有 PINN 的基础上,通过增加 Fourier 层构成 Fourier 神经网络来更好地求解周期边界的微分方程,并且可以预测微分方程的周期解。

但是,激活函数使用正弦或余弦函数时,输入 $\mathbf{x}$ 在 $(2k+1)\pi$ 处的正弦函数梯度为 0,容易出现梯度消失的情况。为了防止这种情况的发生,在 PINN 增加 Fourier 层的同时也使用了残差网络,这样神经网络的输出变为

$$\mathbf{h}^{l+1}(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^m (w_{i,j}^{l+1} \cos(\mathbf{W}^l \mathbf{x}) + v_{i,j}^{l+1} \sin(\mathbf{V}^l \mathbf{x})) + \mathbf{b}^{l+1} + \mathbf{x}. \quad (6)$$

增加残差网络后,使每个网络层中增加了跨层连接,网络能够更好地学习恒等映射,从而有效减缓梯度消失的情况,加快训练速度。网络结构以及残差的 Fourier 层结构如图 3 所示。

损失函数的设置包括三部分:偏微分方程损失、观测数据损失以及边界损失。为了防止神经网络过拟合,损失函数中还加入了 L_1 正则化部分:

$$E = \lambda_d E_d + \lambda_f E_f + \lambda_b E_b + \lambda_l \sum_j |\boldsymbol{\theta}_j|, \quad (7)$$

其中, $\lambda_d, \lambda_f, \lambda_b, \lambda_l$ 是各部分损失所对应的权重,同时能够起到归一化的作用, $|\boldsymbol{\theta}_j|$ 表示神经网络第 j 层的参数矩阵的 L_1 范数。

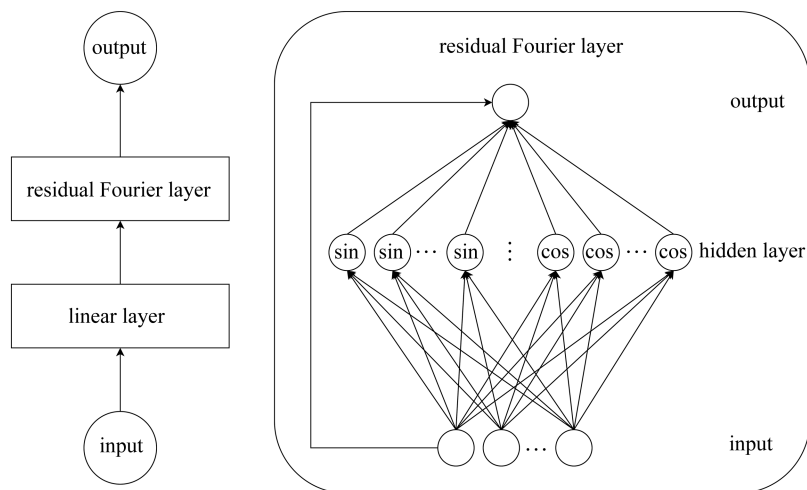


图 3 Res-FNN 结构示意图

Fig. 3 The Res-FNN structure diagram

为了验证模型训练微分方程的有效性,现在仍然选择上述微分方程(4),使用 Res-FNN 进行求解,选用的网络深度与宽度和 PINN 相同,训练次数使用 Adam 优化方法训练 10 000 次,得到的 L_2 误差随区间长度变

化曲线如图4所示。

从图4可以看出,虽然随着区间长度的增加,Res-FNN训练得到的结果与解析解的 L_2 误差总体上仍然会逐渐增加,但是增加的幅度很小,并且 L_2 误差始终保持在 10^{-3} 的一个比较合理的范围内.图5中对比两种网络结构训练的结果,能更清晰地看到残差 Fourier 神经网络处理此问题时具有的优势。

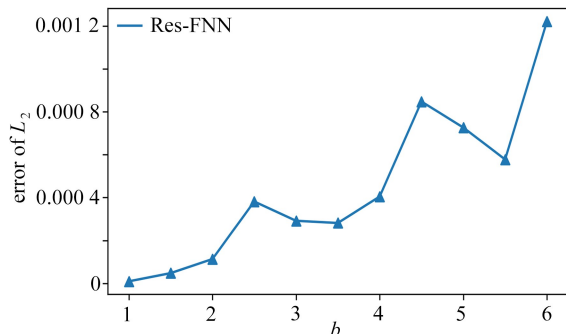


图4 Res-FNN 随区间长度 b 训练误差变化

Fig. 4 The Res-FNN training error variation with interval length b

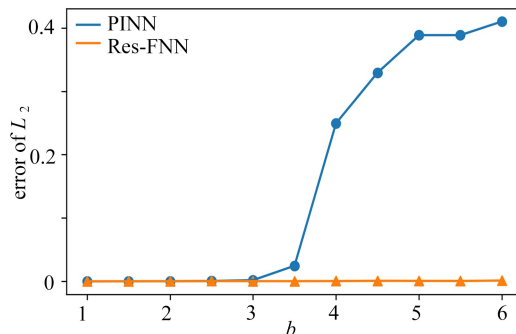


图5 两个不同神经网络 L_2 训练误差随区间长度 b 的变化对比

Fig. 5 Comparison of L_2 training errors with interval length b for 2 different neural networks

图5的结果表明:随着区间长度的增加,Res-FNN的 L_2 误差变化非常小,而且始终保持在一个合理的范围内;而PINN训练结果的 L_2 误差随着区间长度的增加会显著增大,导致无法逼近解析解. Res-FNN会达到这种效果主要跟 $\sin$ 和 $\cos$ 函数的周期性有关,能够保证在较长区间上保持梯度稳定。

为了保证Res-FNN训练结果的有效性,在第3节对线性椭圆型微分方程进行Res-FNN的收敛性分析。

3 收敛性分析

给定Dirichlet边界的二阶线性椭圆型偏微分方程:

$$\begin{cases} \mathcal{L}[u] = f, & u \in U, \\ u = g, & u \in \partial U, \end{cases} \quad (8)$$

其中, $\mathcal{L}[u] = \sum_{i,j=1}^d D_i(a^{ij}(\mathbf{x})D_j u + b^i(\mathbf{x})u) + \sum_{i=1}^d c^i(\mathbf{x})D_i u + d(\mathbf{x})u$.

假设 $N_L(\mathbf{x}; \mathbf{n}, \boldsymbol{\theta}_L)$ 是一个 L 层的神经网络, $\mathbf{n} = (n_0, n_1, \dots, n_L)$ 表示神经网络的第 L 层含有 n_L 个隐藏单元, $\boldsymbol{\theta}_L = (\mathbf{W}^{(L)}, \mathbf{b}^{(L)})$, $\mathbf{W}^{(L)}$ 和 $\mathbf{b}^{(L)}$ 表示第 L 层神经网络中的权重矩阵和偏置.下面定义神经网络类:

$$\mathcal{H}_n^{NN} = \{N_L(\mathbf{x}; \mathbf{n}, \boldsymbol{\theta}_L) : \mathbb{R}^d \mapsto \mathbb{R}^{d_{out}} \mid \boldsymbol{\theta}_L = (\mathbf{W}^{(L)}, \mathbf{b}^{(L)})\}, \quad (9)$$

其中, $N_L(\mathbf{x}; \mathbf{n}, \boldsymbol{\theta}_L)$ 代表的神经网络是Res-FNN,包括线性层和残差 Fourier 层,线性层使用的激活函数为 $\tanh$ 函数,残差 Fourier 层使用的激活函数为 $\sin$ 和 $\cos$ 函数。

Res-FNN优化问题是在一个固定结构的神经网络类上寻找最优的神经网络,相当于寻找最优的神经网络参数使得神经网络损失函数达到最小:

$$\min_{\boldsymbol{\theta}_L} E(N_L(\mathbf{x}; \mathbf{n}, \boldsymbol{\theta}_L); \lambda, \boldsymbol{\lambda}^R).$$

定义损失函数的一个上界,称为Hölder正则化损失函数:

$$E_{\text{Hölder}} = \begin{cases} E(N_L(\mathbf{x}; \mathbf{n}, \boldsymbol{\theta}_L)) + \lambda_{r,m}^R [\mathcal{L}[h]]_{\alpha,U}^2 + \lambda_{b,m}^R [\mathcal{B}[h]]_{\alpha,\partial U}^2, & d \geq 2, \\ E(N_L(\mathbf{x}; \mathbf{n}, \boldsymbol{\theta}_L)) + \lambda_{r,m}^R [\mathcal{L}[h]]_{\alpha,U}^2, & d = 1, \end{cases} \quad (10)$$

其中, $\mathbf{m} = (m_r, m_b)$ 为区域 U 和边界 ∂U 中选取的样本点个数,并且样本点选取的概率分布分别记作 μ_r, μ_b ,常见的概率分布有均匀分布、Gauss分布等. $\boldsymbol{\lambda}_m^R = (\lambda_{r,m}^R, \lambda_{b,m}^R)$ 是惩罚项系数. $[u]_{\alpha,U}$ 是Hölder常数,表达式为

$$[u]_{\alpha,U} = \sup_{x,y \in U; x \neq y} \frac{|u(x) - u(y)|}{\|x - y\|^\alpha} < \infty, \quad 0 < \alpha < 1. \quad (11)$$

令 h_{m_r} 是Hölder正则化损失函数极小时所对应的神经网络,称之为极小化元.由于 $0 \leq E \leq E_{\text{Hölder}}$,证明

Res-FNN 的收敛性相当于证明 $m_r \rightarrow \infty, h_{m_r} \rightarrow u^*$, 在 $C^k(U)$, 其中 m_r 是区域 U 上的训练点个数, $u^* \in \mathcal{H}_n^{NN}$, u^* 是偏微分方程(8)的解. 分三部分进行证明, 首先要证明一个收敛的神经网络序列收敛到给定函数 u , 那么神经网络序列的 k 阶导数也收敛到 u 的 k 阶导数.

定理 1 让 U 是一个有界区域, $\mathcal{H}_n^{NN}$ 是一个具有相同结构 $\mathbf{n} = (n_0, n_1, \dots, n_L)$ 的 Fourier 神经网络类. 对于一个函数 $u \in C^k(\bar{U})$, 让 $\{N_j\}_{j=1}^\infty$ 是 $\mathcal{H}_n^{NN}$ 中的神经网络序列, 并且相关的边界和权重是一致有界的, 在 $C^0(\bar{U})$ 上 $N_j \rightarrow u$. 所以, 在 $C^k(\bar{U})$ 上 $N_j \rightarrow u$.

证明 定理 1 使用数学归纳法进行证明, 具体见附录 A.

为了保证线性椭圆型微分方程的解的存在性、正则性和唯一性, 作出以下假设:

假设 1^[16] 让 $\lambda(\mathbf{x})$ 是 $a^{ij}(\mathbf{x})$ 的最小特征值, 并且 $\alpha \in (0, 1)$,

- ① 对于某个常数 λ_0 , 在 U 中满足 $\lambda(\mathbf{x}) > \lambda_0 > 0$, 并且 $a^{ij} = a^{ji}$;
- ② $c^i(\mathbf{x}), d(\mathbf{x}) \in C^\alpha(U)$;
- ③ a^{ij}, b^{ij} 在 $C^{1,\alpha}(U)$ 上, 并且 $f \in C^{1,\alpha}(U), g \in C^{1,\alpha}(\partial U)$;
- ④ U 在每个边界点都满足外边界条件;
- ⑤ 存在常数 $\Lambda, v \geq 0$ 使得对于所有 $x \in U$, 有

$$\sum_{i,j} |a^{ij}(\mathbf{x})|^2 \leq \Lambda, \lambda_0^{-2}(b(\mathbf{x})^2 + c(\mathbf{x})^2) + \lambda_0^{-1} |d(\mathbf{x})| \leq v^2, \quad (12)$$

其中, $\mathbf{b}(\mathbf{x}) = (b^1(\mathbf{x}), \dots, b^d(\mathbf{x}))^T, c(\mathbf{x}) = (c^1(\mathbf{x}), \dots, c^d(\mathbf{x}))^T$.

然后证明 Res-FNN 求解线性椭圆型偏微分方程在 $C^0(U)$ 上的收敛性定理.

定理 2 假设 1 成立, 并且 $\tilde{c}(\mathbf{x}) = \sum_{i=1}^d D_i b^i(\mathbf{x}) + d(\mathbf{x}) \leq 0$. 设 m_r, m_b 分别为来自概率分布 μ_r, μ_b 的独立同分布(IID)样本的数量, 并且 $m_r = O(m_b^{d/(d-1)})$. 让 $h_{m_r} \in \mathcal{H}_n^{NN}$ 是 Hölder 正则化损失函数的极小化元, 那么以下情况成立:

- ① 对于满足假设 1 的每一个偏微分方程数据集 $\{f, g\}$, 偏微分方程(8)存在唯一经典解 u^* ;
- ② 对于 IID 样本概率为 1, 则

$$\lim_{m_r \rightarrow \infty} h_{m_r} = u^*, \quad u^* \in C^0(U).$$

证明 详细证明参见文献[6].

最后结合定理 1, 将定理 2 的在 $C^0(U)$ 上的收敛性推广到 $C^k(U)$ 上, 得到 Res-FNN 求解二阶线性椭圆型偏微分方程的收敛性定理.

定理 3 假设 $\mathcal{H}_n^{NN}$ 是一个具有相同结构 $\mathbf{n} = (n_0, n_1, \dots, n_L)$ 的残差 Fourier 神经网络类, 使得 $u^* \in \mathcal{H}_n^{NN}$, 其中 u^* 是偏微分方程(8)的解. 对于所有 m_r , 令 $\mathcal{H}_{m_r} = \mathcal{H}_n^{NN}$. 对于 IID 样本上的概率为 1, 有

$$\lim_{m_r \rightarrow \infty} h_{m_r} = u^*, \quad u^* \in C^k(U).$$

证明过程见附录 B.

这个定理表明了, 在满足上述假设并且训练点足够多的情况下, 残差 Fourier 神经网络最小化损失函数的极小化元可以收敛到线性椭圆微分方程的解. 第 4 节的数值实验中将对比 PINN、FNN 和 Res-FNN 三种不同的神经网络结构求解线性椭圆型偏微分方程的误差以及损失函数的下降速度, 然后使用 Res-FNN 求解反问题并进行鲁棒性测试.

4 数值实验

在本节中, 考虑到隐藏层层数及隐藏层单元个数不同会影响训练结果, 我们使用具有 7 个隐藏层, 每层 120 个隐藏单元的 PINN、FNN 和 Res-FNN 三种神经网络进行训练, 线性层的激活函数选择 tanh 函数. 优化过程采用 Adam 算法先进行训练, 再使用 L-BFGS 算法进行精调. 所有计算均在 NVIDIA GeForce RTX 4060 Laptop GPU (8GB) 上完成.

4.1 微分方程正问题

Poisson 方程:

$$\begin{cases} \Delta u = \sin(\pi x) \sin(\pi y), \\ u(-2, y) = u(2, y) = 0, \\ u(x, -2) = u(x, 2) = 0. \end{cases} \quad (13)$$

此微分方程有解析解

$$u(x, y) = \frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y).$$

选取 Adam 算法初始学习率为 0.001, L-BFGS 算法初始学习率为 0.8, 首先使用 Adam 优化器训练 500 次, 比较 PINN、FNN 和 Res-FNN 三种模型求解 Poisson 方程的损失函数下降速度.

从图 6 中可以看到训练次数较少时, FNN 的损失函数值较小, 随着训练次数的增加, Res-FNN 的训练速度大于 FNN 和 PINN 的训练速度, 从而在相同训练次数时损失函数值达到三种模型中的最小值, 说明了在 Fourier 层中加入残差以后会提高训练速度. 下面比较结构相同的三种模型在相同训练次数下的求解精度.

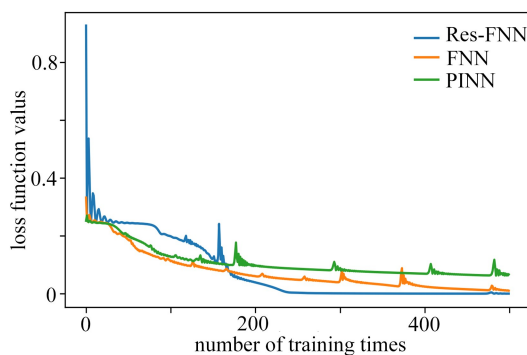


图 6 三种方法损失函数值的下降曲线

Fig. 6 The decline curves of loss function values for 3 methods

图 7 展示了 PINN、FNN 和 Res-FNN 三种神经网络求解上述 Poisson 方程得到的解和误差. 初始配置点在 $[-2, 2]$ 的正方形区域等距选取 1 600 个点.

表 1 表明了相同网络深度和宽度以及相同训练次数的情况下, Res-FNN 在 L_∞ 范数和 L_2 范数下的误差可以取得最高的精度. FNN 和 Res-FNN 在 L_∞ 范数下的误差可以达到 10^{-5} 级别, 而 PINN 只达到 10^{-4} 级别, FNN 在 L_2 范数下的误差也小于 PINN, 说明 Fourier 神经网络在求解周期边界微分方程上与 PINN 相比具有优势, 并且 Res-FNN 的训练精度也要高于 FNN.

表 1 不同神经网络模型训练结果的 L_∞ 误差和 L_2 误差及其对应训练次数

Table 1 The training result errors of L_∞ and L_2 for different neural network models, as well as the corresponding numbers of training times

model	error of L_∞	error of L_2	number of training times (Adam+L-BFGS)
Res-FNN	5.757×10^{-5}	1.091×10^{-5}	5 000+1 500
FNN	6.184×10^{-5}	1.950×10^{-5}	5 000+1 500
PINN	1.627×10^{-4}	3.475×10^{-5}	5 000+1 500

Helmholtz 方程:

$$\begin{cases} \Delta u + 2u - f(x, y) = 0, \\ u(-2, y) = u(2, y) = 0, \\ u(x, -2) = u(x, 2) = 0, \end{cases} \quad (14)$$

其中, 源项 $f(x, y)$ 为

$$f(x, y) = -\pi^2 \sin(\pi x) \sin(2\pi y) - 4\pi^2 \sin(\pi x) \sin(2\pi y) + 2\sin(\pi x) \sin(2\pi y),$$

该方程解析解为

$$u(x, y) = \sin(\pi x) \sin(2\pi y).$$

选取 Adam 算法初始学习率为 0.001, L-BFGS 算法初始学习率为 0.5, 首先使用 Adam 优化器训练 2 000 次, 比较 PINN、FNN 和 Res-FNN 三种模型求解 Helmholtz 方程的损失函数下降速度.

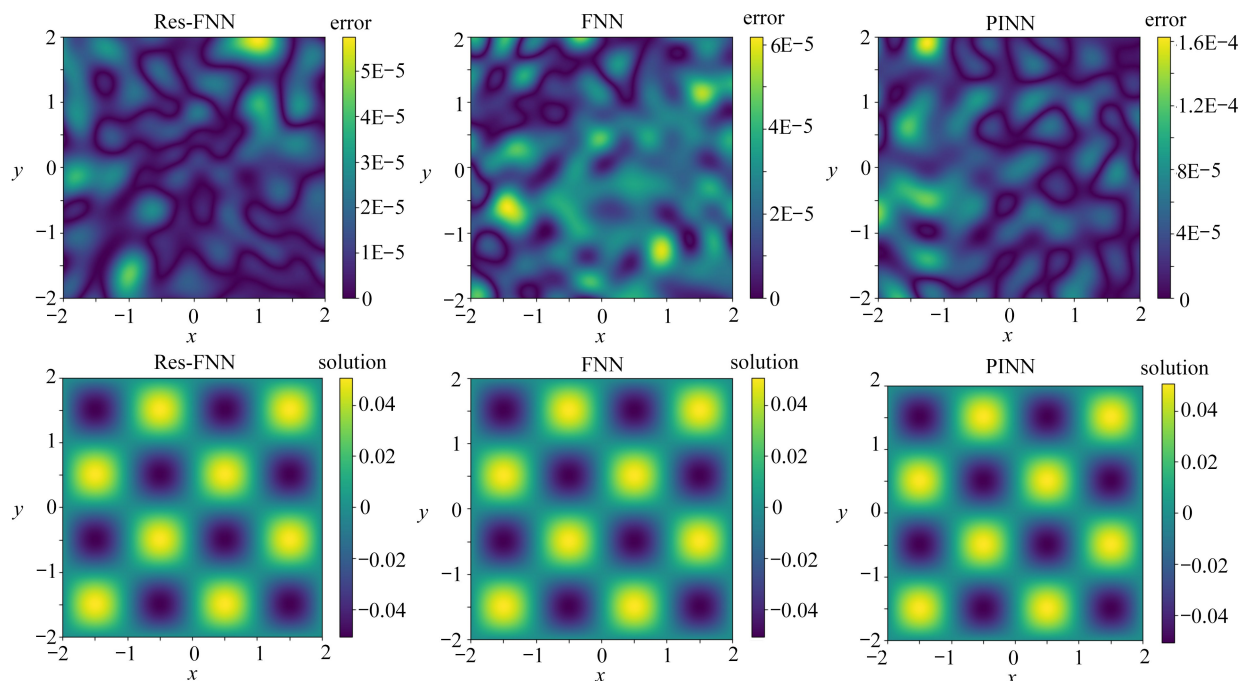


图 7 Res-FNN、FNN 和 PINN 求解 Poisson 方程的结果

Fig. 7 The results of solving Poisson equation with the Res-FNN, the FNN and the PINN

从图 8 中可以看出, 三种模型在训练开始的损失函数下降速度都比较小, 随着训练次数的增加, Res-FNN 的损失函数值以最快的下降速度下降, 然后是 FNN, 最后是 PINN. 求解 Helmholtz 方程与求解 Poisson 方程时的损失函数值曲线的趋势基本相同, 验证了 Res-FNN 在求解线性椭圆型偏微分方程时具有训练速度快的优势.

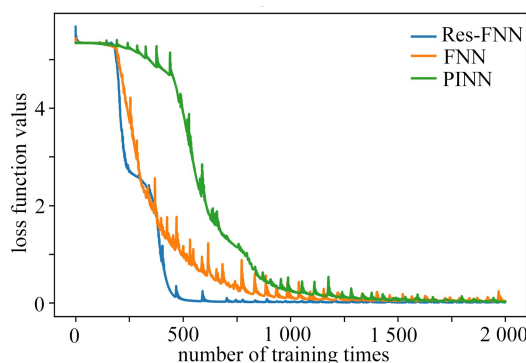


图 8 三种方法损失函数值的下降曲线

Fig. 8 The decline curves of loss function values for 3 methods

图 9 展示了 PINN、FNN、Res-FNN 三种神经网络求解 Helmholtz 方程得到的解的误差. 三种神经网络的网络结构与上面保持一致, 初始配置点在 $[-2, 2]$ 的正方形区域等距选取 1 600 个点.

表 2 表明了同样在相同网络深度和宽度以及相同训练次数的情况下, Res-FNN 在 L_∞ 范数和 L_2 范数下的误差最小, 分别是 4.110×10^{-3} 和 1.140×10^{-3} , 然后 FNN 的误差同样也小于 PINN 的误差, 说明 Res-FNN 求解线性椭圆型偏微分方程时具有求解精度高的优势.

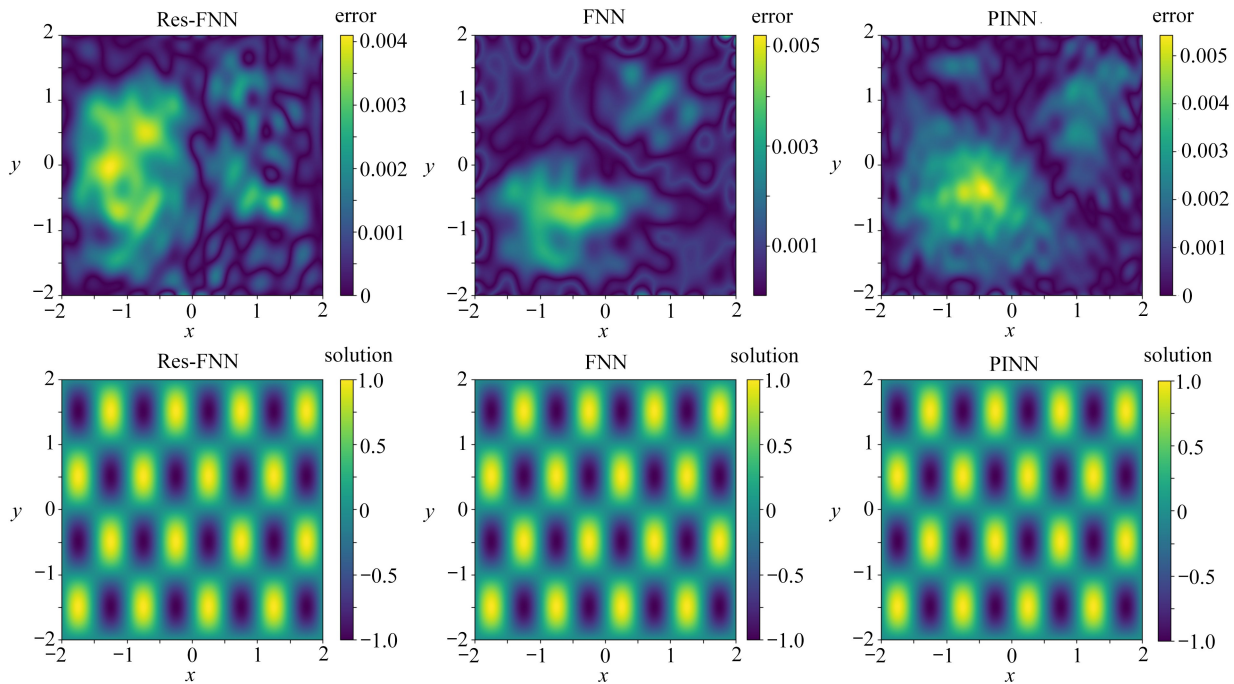


图9 Res-FNN、FNN 和 PINN 求解 Helmholtz 方程的结果

Fig. 9 The results of solving the Helmholtz equation with Res-FNN, FNN and PINN

表2 不同神经网络模型训练结果的 L_∞ 误差和 L_2 误差及其对应训练次数Table 2 The training result errors of L_∞ and L_2 for different neural network models, as well as the corresponding numbers of training times

model	error of L_∞	error of L_2	number of training times (Adam+L-BFGS)
Res-FNN	4.110×10^{-3}	1.140×10^{-3}	10 000+25 000
FNN	5.234×10^{-3}	1.204×10^{-3}	10 000+25 000
PINN	5.434×10^{-3}	1.221×10^{-3}	10 000+25 000

4.2 微分方程反问题

使用 Res-FNN 解决微分方程反问题时,在整个区域内等距离选取 1 000 个点作为神经网络的训练点,然后使用 Adam 和 L-BFGS 优化器进行训练.图 10 和图 11 展示了 Poisson 方程和 Helmholtz 方程训练点的选取以及训练后的结果,图中黑色点表示选取的训练点.网络结构选择具有 7 层,每层 120 个隐藏单元的 Res-FNN.训练时,对训练数据增加 1% 的噪声,来判断神经网络训练结果的有效性.

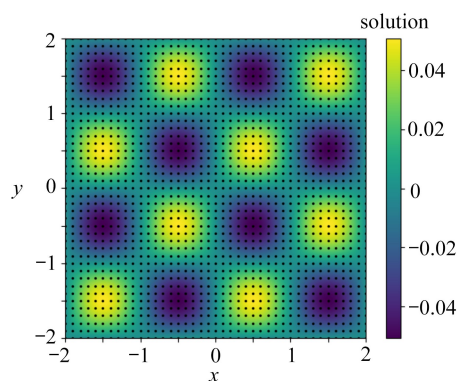


图10 Res-FNN 求解 Poisson 方程反问题的结果

Fig. 10 The results of solving the inverse problem of the Poisson equation with the Res-FNN

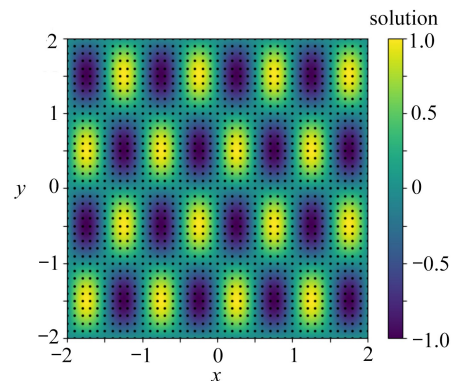


图11 Res-FNN 求解 Helmholtz 方程反问题的结果

Fig. 11 The results of solving the inverse problem of the Helmholtz equation with the Res-FNN

表 3 中使用 Res-FNN 训练 Poisson 方程反问题时,在无噪声的情况下,参数 λ 训练后的结果为 1.000 05;增加 1%的噪声以后训练结果为 1.000 54.使用 Res-FNN 求解 Helmholtz 方程反问题时参数反演的结果,在无噪声的情况下参数 λ 在神经网络训练后的结果为 2.003 34;在原始数据增加 1%噪声的情况下, λ 在神经网络训练后的结果为 2.009 93.为了测试模型的鲁棒性,选取不同强度的噪声修改训练数据,计算 Res-FNN 训练反问题结果的相对误差.

从表 4 中可以看出,即使噪声强度增加到 10%,Res-FNN 训练得到的 Poisson 方程反问题的参数相对误差是 0.664%;Res-FNN 求解 Helmholtz 方程反问题时,无噪声干扰时的相对误差达到 0.167%,但与求解 Poisson 方程反问题相比,随着噪声强度的提高,Helmholtz 方程反问题相对误差的增长速度明显快于 Poisson 方程,当噪声强度达到 10%时,相对误差是 4.216%,仍然处在一个合理的范围内,说明 Res-FNN 具有良好的鲁棒性,可以用来进行微分方程反问题的快速反演计算.

表 3 Res-FNN 求解 Poisson 方程和 Helmholtz 方程在干净数据和 1%噪声数据下反问题结果

Table 3 Res-FNN solutions to the inverse problems of the Poisson equation and the Helmholtz equation on clean data and 1% noisy datas

correct PDE	identified PDE (clean data)	identified PDE (1% noise)
$\Delta u = \sin(\pi x) \sin(\pi y)$	$\Delta u = 1.000\ 05 \sin(\pi x) \sin(\pi y)$	$\Delta u = 1.000\ 54 \sin(\pi x) \sin(\pi y)$
$\Delta u + \lambda u = f(x, y)$	$\Delta u + 2.003\ 34 u = f(x, y)$	$\Delta u + 2.009\ 93 u = f(x, y)$

表 4 Res-FNN 求解 Poisson 方程和 Helmholtz 方程在不同噪声强度下的反问题结果

Table 4 The inverse problem results of solving the Poisson equation and the Helmholtz equation with the Res-FNN under different noise intensities

noise/%	relative error of the Poisson equation/%	relative error of the Helmholtz equation/%
0	0.005	0.167
1	0.054	0.497
5	0.105	2.976
10	0.664	4.216

5 结 论

本文提出了 Res-FNN 用于解决 PINN 在广域问题上求解精度低、训练速度慢的问题,并且对线性椭圆型偏微分方程进行了收敛性分析.数值实验表明 Res-FNN 求解线性椭圆型偏微分方程正问题与 PINN 和 FNN 对比具有收敛速度快,求解精度高的优势;Res-FNN 求解反问题也具有良好的鲁棒性,但微分方程的源项会影响反问题结果的精度.然而,Res-FNN 也具有局限性:Res-FNN 适用于周期边界的微分方程,在处理非周期边界的微分方程时的精度较低;并且由于采用的是截断的 Fourier 变换,会导致高频信息丢失.在下一步研究中,可将收敛性定理扩展到非线性椭圆型微分方程,并通过数值实验求解非线性椭圆型微分方程.

附 录 A

定理 1 证明 令 $\mathbf{n} = (n_1, n_2, \dots, n_L)$ 是 Fourier 神经网络的结构, Fourier 神经网络的每一层的输出可以表示为

$$\mathbf{h}^{l+1}(\mathbf{x}) = \begin{cases} \mathbf{W}^{l+1} \sigma(\mathbf{h}^l(\mathbf{x})) + \mathbf{b}^{l+1}, & 0 \leq l < L-1, \\ \sum_{i=1}^m \sum_{j=1}^m (w_{i,j}^{l+1} \cos(\mathbf{W}^l \mathbf{h}^l(\mathbf{x})) + v_{i,j}^{l+1} \sin(\mathbf{V}^l \mathbf{h}^l(\mathbf{x}))) + \mathbf{b}^{l+1} + \mathbf{h}^l(\mathbf{x}), & l = L-1, \end{cases} \quad (\text{A1})$$

其中, $\mathbf{h}^0(\mathbf{x}) = \mathbf{x}$, 令 $\boldsymbol{\theta}_F = (\mathbf{W}^l, \mathbf{b}^l, \mathbf{V}^{L-1})_{l=0}^{L-1} \in \mathcal{A}_n^{NN}$ 是一个具有相同结构的 Fourier 神经网络类, 并且 $\boldsymbol{\theta}_F$ 是一致有界的, 所以存在收敛子列记为 $\boldsymbol{\theta}_F^i$, 收敛到 $\boldsymbol{\theta}_F^*$. $\boldsymbol{\theta}_F^i$ 对应的神经网络记为 N_F^i , $\boldsymbol{\theta}_F^*$ 对应的神经网络记为 N_F^* .

首先要证明对于所有 $l = 0, 1, \dots, L-1$, $D^s \mathbf{h}^{l+1}(\mathbf{x}; \boldsymbol{\theta}_F^i)$ 在 $C^0(U)$ 上收敛到 $D^s \mathbf{h}^{l+1}(\mathbf{x}; \boldsymbol{\theta}_F^*)$, $\forall s, 0 \leq |s| \leq k$. 由于前 $L-1$ 层是经典的 PINN, 在文献[6]已经使用数学归纳法进行了详细的证明, 因此本文中只需要证明在第 L 层——残差 Fourier 层上: $D^s \mathbf{h}^L(\mathbf{x}; \boldsymbol{\theta}_F^i)$ 在 $C^0(U)$ 上收敛到 $D^s \mathbf{h}^L(\mathbf{x}; \boldsymbol{\theta}_F^*)$, $\forall s, 0 \leq |s| \leq k$, 即可.

设 h_k^l 是第 l 层输出向量的第 k 个输出, $\sum_{i=1}^m \sum_{j=1}^m (w_{i,j}^L \cos(\mathbf{W}^L \mathbf{h}^L(\mathbf{x})) + v_{i,j}^L \sin(\mathbf{V}^L \mathbf{h}^L(\mathbf{x}))) + \mathbf{b}^{L+1} + \mathbf{h}^L(\mathbf{x})$ 可以写为

$W^L \cos(W^{L-1} h^{L-1}(x)) + V^L \cos(V^{L-1} h^{L-1}(x)) + b^{L+1} + h^L(x)$. 由公式

$$D^s h_k^L(x; \theta_{F,L}) = \sum_{i=1}^{n_L} W_{ij}^{L-1} \sum_{t=0}^{|s|} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x) \sigma^{(t)}(h_k^{L-1}(x; \theta_{F,L-1})), \quad (A2)$$

可以得到在第 L 层的残差 Fourier 神经网络层

$$D^s h_k^L(x; \theta_{F,L}) = \sum_{j=1}^{n_{L-1}} W_{ij}^{L-1} \sum_{i=1}^{n_L} W_{ii}^{L-1} \sum_{t=0}^{|s|} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x) \cos^{(t)}(h_k^{L-1}(x; \theta_{F,L-1})) + \sum_{j=1}^{n_{L-1}} V_{ij}^{L-1} \sum_{i=1}^{n_L} V_{ii}^{L-1} \sum_{t=0}^{|s|} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x) \sin^{(t)}(h_k^{L-1}(x; \theta_{F,L-1})) + D^s h_k^{L-1}(x; \theta_{F,L-1}). \quad (A3)$$

其中

$$Q_{i,t}^{L-1}(\theta_{F,L-1}, s, x) = \begin{cases} Q_{i,s-1}^{L-1}(\theta_{F,L-1}, s-1, x) \frac{\partial h_i^{L-1}(x; \theta_{F,L-1})}{\partial \tilde{x}_s}, & t = s, \\ \frac{\partial}{\partial \tilde{x}_s} Q_{i,t}^{L-1}(\theta_{F,L-1}, s-1, x) + Q_{i,t-1}^{L-1}(\theta_{F,L-1}, s-1, x) \frac{\partial h_i^{L-1}(x; \theta_{F,L-1})}{\partial \tilde{x}_s}, & 1 < t < s, \\ \frac{\partial}{\partial \tilde{x}_s} Q_{i,1}^{L-1}(\theta_{F,L-1}, s-1, x), & t = 1, \\ 0, & t = 0. \end{cases}$$

通过上面的公式,我们就得到了一个相邻隐藏层之间的递推公式,使用这个公式来证明收敛性.

$D^s h_k^L(x; \theta_{F,L})$ 由三部分组成, $D^s h_k^{L-1}(x; \theta_{F,L-1})$ 已经被证明,其余两部分的不同只有激活函数的不同,所以只需要证明其中一部分的收敛性.

令

$$D^s h_{k,\cos}^L(x; \theta_{F,L}) = \sum_{j=1}^{n_{L-1}} W_{ij}^{L-1} \sum_{i=1}^{n_L} W_{ii}^{L-1} \sum_{t=0}^{|s|} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x) \cos^{(t)}(h_k^{L-1}(x; \theta_{F,L-1})).$$

那么利用三角不等式:

$$|D^s h_{k,\cos}^L(x; \theta_{F,L}) - D^s h_{k,\cos}^L(x; \theta'_{F,L})| \leq \sum_{j=1}^{n_{L-1}} \sum_{i=1}^{n_L} \sum_{t=0}^{|s|} |W_{ij}^{L-1} W_{ii}^{L-1} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x) - W_{ij}^{L-1} W_{ii}^{L-1} Q_{i,t}^{L-1}(\theta'_{F,L-1}, |s|, x)| |\cos^{(t)}(h_k^{L-1}(x; \theta_{F,L-1}))| + \sum_{j=1}^{n_{L-1}} \sum_{i=1}^{n_L} \sum_{t=0}^{|s|} |W_{ij}^{L-1} W_{ii}^{L-1} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x)| |\cos^{(t)}(h_k^{L-1}(x; \theta_{F,L-1})) - \cos^{(t)}(h_k^{L-1}(x; \theta'_{F,L-1}))|,$$

$|W_{ij}^{L-1} W_{ii}^{L-1} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x) - W_{ij}^{L-1} W_{ii}^{L-1} Q_{i,t}^{L-1}(\theta'_{F,L-1}, |s|, x)|$ 由归纳假设可以证明收敛, $|\cos^{(t)}(h_k^{L-1}(x; \theta_{F,L-1}))| \leq 1$, 又因为 $|W_{ij}^{L-1} W_{ii}^{L-1} Q_{i,t}^{L-1}(\theta_{F,L-1}, |s|, x)| \leq M$ 有界, 并且 $\cos^{(t)}$ 是连续函数, 所以

$$\lim_{\theta_{F,L-1} \rightarrow \theta'_{F,L-1}} \cos^{(t)}(h_k^{L-1}(x; \theta_{F,L-1})) - \cos^{(t)}(h_k^{L-1}(x; \theta'_{F,L-1})) = 0.$$

因此

$$\lim_{\theta_{F,L-1} \rightarrow \theta'_{F,L-1}} |D^s h_{k,\cos}^L(x; \theta_{F,L}) - D^s h_{k,\cos}^L(x; \theta'_{F,L})| = 0.$$

所以对于收敛子列 $\theta_{F,i}^j$ 有

$$\lim_{\theta_{F,i}^j \rightarrow \theta_F^*} |D^s h^L(x; \theta_{F,i}^j) - D^s h^L(x; \theta_F^*)| = 0.$$

又因为子列的选取是任意的, 所以定理 1 成立.

附 录 B

定理 3 证明 由定理 2 可以得到: 当 $m_r \rightarrow \infty$ 时, $\|h_{m_r} - u^*\|_{C^0(U)} \rightarrow 0$, 根据定理 1 可以得到: $m_r \rightarrow \infty$ 时, $\|h_{m_r} - u^*\|_{C^k(U)} \rightarrow 0$. 证毕.

参考文献 (References):

- [1] 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.
- [2] Cai S, Mao Z, Wang Z. Physics-informed neural networks (PINNs) for fluid mechanics: a review[J]. *Acta Mechanica Sinica*, 2021, **37**(12): 1727-1738.
- [3] Mao Z, Jagtap A D, Karniadakis G E. Physics-informed neural networks for high-speed flows[J]. *Computer Methods in Applied Mechanics and Engineering*, 2020, **360**: 112789.
- [4] Jin X, Cai S, Li H. NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations[J]. *Journal of Computational Physics*, 2021, **426**: 109951.
- [5] Arzani A, Wang J X, D'souza R M. Uncovering near-wall blood flow from sparse data with physics-informed neural networks[J]. *Physics of Fluids*, 2021, **33**(7): 071905.
- [6] Shin Y, Darbon J, Karniadakis G E. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs[J]. *Communications in Computational Physics*, 2020, **28**(5): 2042-2074.
- [7] Kharazmi E, Zhang Z, Karniadakis G E. Hp-VPINNs: variational physics-informed neural networks with domain decomposition[J]. *Computer Methods in Applied Mechanics and Engineering*, 2021, **374**: 113547.
- [8] Jagtap A D, Karniadakis G E. Extended physics-informed neural networks (XPINNs): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations[J]. *Communications in Computational Physics*, 2020, **28**(5).
- [9] 闵建, 傅卓佳, 郭远. 课程-迁移学习物理信息神经网络用于曲面长时间对流扩散行为模拟[J]. 应用数学和力学, 2024, **45**(9): 1212-1223. (Min Jian, Fu Zhuojia, Guo Yuan. Curriculum-transfer-learning-based physics-informed neural networks for simulating long-term-evolution convection-diffusion behaviors on curved surfaces [J]. *Applied Mathematics and Mechanics*, 2024, **45**(9): 1212-1223. (in Chinese))
- [10] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. *Neural networks*, 1989, **2**(5): 359-366.
- [11] Liu Z, Wang Y, Vaidya S. KAN: Kolmogorov-arnold networks[PP/OL]. V5. (2025-02-09). <https://arxiv.org/abs/2404.19756>.
- [12] Kolmogorov A N. On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables[J]. *Doklady Akademii Nauk SSSR*, 1956, **108**(2): 369-373. (in Russian)
- [13] Liu Shuang. Fourier neural network for machine learning[C]//2013 International Conference on Machine Learning and Cybernetics. Tianjin, 2014: 285-290.
- [14] Ngom M, Marin O. Fourier neural networks as function approximators and differential equation solvers[J]. *Statistical Analysis and Data Mining: the ASA Data Science Journal*, 2021, **14**(6): 647-661.
- [15] Rahaman N, Baratin A, Arpit D. On the spectral bias of neural networks[C]//Proceedings of the 36th International Conference on Machine Learning, 2019: 5301-5310.
- [16] Gilbarg D, Trudinger N S. *Elliptic Partial Differential Equations of Second Order*[M]. Berlin, Heidelberg: Springer, 1977.