



# 基于残差神经网络模型的 Fredholm 积分方程数值解法

张殿焜, 靳聪明

(浙江理工大学理学院, 杭州 310018)

**摘要:** 为了求解 Fredholm 积分方程, 特别是高维 Fredholm 积分方程, 提出了一种采用残差神经网络求解 Fredholm 积分方程的数值方法。首先在求解区域随机产生训练数据集, 通过前向传播残差神经网络得到训练集上的预测值; 然后代入 Fredholm 积分方程得到离散格式, 并定义损失函数, 将解 Fredholm 积分方程转化为一个最小二乘问题; 最后利用残差神经网络进行优化求解。该方法形式简单, 对高维 Fredholm 积分方程求解问题计算量无显著增加。数值实验表明: 该方法能有效求解 Fredholm 积分方程, 且能取得很好的收敛精度; 所训练的残差神经网络不会出现网络退化现象, 表现出稳定性好、泛化能力强等优点。

**关键词:** 残差神经网络; Fredholm 积分方程; 高维积分方程; 最小二乘法

中图分类号: O175.5; TP183

文献标志码: A

文章编号: 1673-3851(2020)05-0706-08

## Numerical method of Fredholm integral equation based on residual neural network model

ZHANG Diankun, JIN Congming

(School of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** To solve Fredholm integral equations, especially high-dimensional Fredholm integral equations, a numerical method is proposed based on residual neural networks. Firstly, the training dataset was generated at random in the solution domain, and the predicted value on the training set was gained by forward propagation of residual neural network. Then, the discretization scheme was obtained by substituting the predicted value into the Fredholm integral equation, and the loss function was defined. Then, solving Fredholm integral equation was transformed into a least squares problem. Finally, residual neural network was used for optimizing the solution. The new method has a simple form and does not significantly increase the computational test for high-dimensional Fredholm integral equation problems. Numerical experimental results show that the new method can solve Fredholm integral equations efficiently and accurately and get good convergence precision. The residual neural networks will not suffer from network degradation, and has the advantages of good stability and strong generalization ability.

**Key words:** residual neural networks; Fredholm integral equation; high dimensional integral equation; least squares method

## 0 引言

1943 年, 心理学家 McCulloch 和数学逻辑专家 Pitts 建立了神经元的数学模型(M-P 模型), 奠定了神经网络模型的基础<sup>[1]</sup>。20 世纪, 由于计算机计算能力的限制和高效训练算法的缺乏, 神经网络并没有表现出

收稿日期: 2020-01-10 网络出版日期: 2020-05-08

基金项目: 国家自然科学基金项目(11571314)

作者简介: 张殿焜(1993—), 男, 河南平顶山人, 硕士研究生, 主要从事机器学习方面的研究。

通信作者: 靳聪明, E-mail: jincm@lsec.cc.ac.cn

优异的学习能力。直到 2006 年,大量已标注数据为神经网络的训练提供了数据准备,GPU(图形处理器)为深度神经网络的训练提供了高速硬件设备,Hinton<sup>[2]</sup>提出的贪婪逐层预训练策略为深度神经网络的训练提供了有效的训练方法,从而掀起了深度神经网络在工业界和学术界的研究热潮,并在人工智能领域的计算机视觉<sup>[3]</sup>、自然语言处理<sup>[4]</sup>、情感分析<sup>[5-7]</sup>等高维问题中取得了巨大的成功。深度神经网络为高维数学问题提供了新思路、新工具,表现出巨大的潜力。

深度神经网络在数值求解微分方程领域已经有了一些新进展。Yi 等<sup>[8]</sup>将神经网络模型应用于求解 Lyapunov 矩阵方程。E 等<sup>[9-11]</sup>将深度神经网络模型用于求解高维椭圆型偏微分方程、高维抛物型偏微分方程和向后随机微分方程。Ray 等<sup>[12]</sup>利用神经网络模型确定微分方程解的不连续区域,用于优化区域剖分。

积分方程作为近代数学的一个重要分支,与微分方程、算子理论等都有紧密的联系<sup>[13]</sup>。由于大部分积分方程解析解很难求得,因此必须考虑如何获得积分方程的数值解。Fredholm 积分方程是一类重要的积分方程,很多学者提出了不同的数值方法,如泰勒多项式逼近法<sup>[14]</sup>、小波分析法<sup>[15]</sup>、Galerkin 投影法<sup>[16]</sup>、最大熵方法<sup>[17]</sup>、Nyström 方法<sup>[18]</sup>、迭代法<sup>[19-20]</sup>等。目前提出的这些数值方法一般都是针对一维和二维积分方程,由于随着积分方程维数增长计算量增长太快,出现维度灾难,很难用于求解高维 Fredholm 积分方程。但在物理、金融等领域经常会遇到各种高维积分方程,因此需要寻找更有效的近似方法。

深度神经网络模型在求解高维问题方面已有很好的表现,基于此本文提出了一种残差神经网络求解 Fredholm 积分方程的数值方法。该方法首先通过随机产生的数据集离散 Fredholm 积分方程;然后在此基础上定义损失函数,通过训练残差神经网络极小化损失函数,得到 Fredholm 积分方程的近似解。

## 1 残差神经网络模型

神经网络是由大量神经元相互连接而形成的网络,神经网络架构示意图如图 1 所示。

神经网络由输入层、隐藏层和输出层构成,图 1 中是一个 4 层的神经网络,LayerL<sub>1</sub> 为输入层,LayerL<sub>2</sub> 和 LayerL<sub>3</sub> 为隐藏层,LayerL<sub>4</sub> 为输出层。其中:  $x = (t_1, t_2)$  是神经网络的输入;  $b_j$  ( $j=1,2,3$ ) 表示第  $j$  层的阈值;  $w_{i,j}$  ( $i=1,2,\dots,6, j=3,4,\dots,7$ ) 表示神经元  $i$  和  $j$  之间的连接权重;  $a_i$  为第  $i$  个神经元各个输入的加权和;  $z_i$  为第  $i$  个神经元的输出(作为下一层的输入);  $F(x, \theta)$  是整个神经网络的输出,也是神经网络的预测值,  $\theta$  表示所有

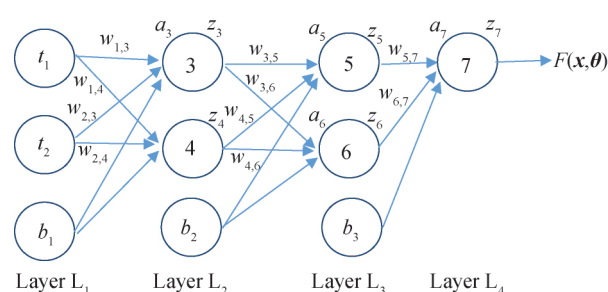


图 1 神经网络架构示意图

权重  $w_{i,j}$  和阈值  $b_j$  的参数向量,可以通过随机赋值进行初始化。神经元的输出由一非线性变换给出,例如神经元 5 的输出为  $z_5 = \varphi(a_5) = \varphi(z_3 \cdot w_{3,5} + z_4 \cdot w_{4,5} + b_2)$ ,其中  $\varphi(\cdot)$  为激活函数,本文使用的激活函数为 ReLU 函数,即  $\varphi(t) = \max\{0, t\}$ 。

设训练集  $\varphi = \{(x_1, Y_1), (x_2, Y_2), \dots, (x_n, Y_n)\}$  有  $n$  对数据,其中  $Y_i$  为  $x_i$  处的观测值。输入数据  $x_i$  通过输入层→隐藏层→输出层的过程称为前向传播(Forward propagation)。通过前向传播获得的预测值和观测值之间一般会存在差异,通常定义一个损失函数(Loss function)

$$\text{loss}(\theta; x_1, x_2, \dots, x_n, Y_1, Y_2, \dots, Y_n) = \frac{1}{2} \sum_{i=1}^n (F(x_i, \theta) - Y_i)^2 \quad (1)$$

用于衡量机器学习模型的预测能力,学习的目标就是使这个损失函数尽可能的小。通过链式法则从输出层到输入层逐层表示出损失函数关于参数  $\theta$  的梯度方向,基于梯度下降法更新参数  $\theta$ ,这一过程称为误差的反向传播(Back propagation)。前向传播和反向传播交替进行,不断迭代调整参数  $\theta$ ,直至网络收敛到稳定状态。图 1 这种含多个隐藏层的神经网络称为深度神经网络。关于深度神经网络模型相关概念及性质详见文献<sup>[21]</sup>。

深度神经网络遇到的最大挑战是随着网络加深,极小化损失函数的过程就会越难,会出现梯度消失和梯度爆炸等问题。一般的应对方法是寻找更好的激活函数或者数据正则化,这样梯度消失的问题得以有效控制,但随着网络深度加深,又出现了错误率先降低后上升的网络退化问题。残差神经网络的提出不仅可以解决梯度消失问题,还可以解决网络退化问题<sup>[22]</sup>。深度残差神经网络的基本组成部分是残差块,其结构示意

图如图2所示。

图2中的残差块有两层隐藏层。第一个隐藏层的输入为 $x$ ,它的输出作为第二隐藏层的输入,就像一般的神经网络。第二个隐藏层的输出与残差块输入 $x$ 的加和作为整个残差块的输出数据,即 $H(x)=\varphi(x)+x$ (如果是卷积神经网络,卷积层输出和输入元素维度必须相同)。在第二层输出值加上 $x$ ,这条路径称作残差连接。将图2这种残差块一个一个串联起来,就得到了残差神经网络。

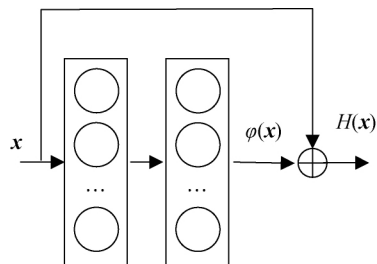


图2 残差块结构示意图

## 2 残差神经网络求解 Fredholm 积分方程

Fredholm 方程第二型积分方程<sup>[13]</sup>的基本形式为:

$$f(x) - \int_{\Omega} k(x, y) f(y) dy = g(x) \quad (2)$$

其中:  $x, y \in \Omega \subset \mathbf{R}^m$ ,  $k(x, y)$  称为积分方程的核函数,  $g(x)$  为自由项。 $g(x)$  和  $k(x, y)$  均已知, 而  $f(x)$  为未知函数。

下面用残差神经网络去学习  $f(x)$ 。对任意一点  $x \in \Omega$ , 设神经网络的输出  $F(x, \theta)$  是  $f(x)$  的近似。在积分区域  $\Omega$  上随机取若干个训练点  $\{x_1, x_2, \dots, x_n\}$  作为训练集。把近似解  $F(x_i, \theta)$  代入式(2)并在训练集上进行离散, 可得到:

$$F(x_i, \theta) - \int_{\Omega} k(x_i, y) F(y, \theta) dy = g(x_i), \quad i = 1, 2, \dots, n \quad (3)$$

其中积分项  $\int_{\Omega} k(x_i, y) F(y, \theta) dy$  可以利用蒙特卡罗方法进行数值积分, 即:

$$\int_{\Omega} k(x_i, y) F(y, \theta) dy \approx \frac{\beta}{n} \sum_{j=1}^n k(x_i, x_j) F(x_j, \theta) \quad (4)$$

其中  $\beta = \int_{\Omega} dx$  表示积分区域的体积。接下来定义损失函数:

$$loss_x(\theta) = \frac{1}{n} \sum_{i=1}^n \left( F(x_i, \theta) - \frac{\beta}{n} \sum_{j=1}^n k(x_i, x_j) F(x_j, \theta) - g(x_i) \right)^2 \quad (5)$$

用残差神经网络极小化损失函数训练参数  $\theta$ , 即求解如式(6)的最小二乘问题。

$$\min_{\theta} loss_x(\theta) \quad (6)$$

训练神经网络需要大量的数据, 在计算过程中可以在积分区域  $\Omega$  上多次随机产生训练样本  $\{x_i\}_{i=1}^n$ , 对参数  $\theta$  进行训练, 从而得到 Fredholm 积分方程的近似解。

为了衡量所得残差神经网络模型的泛化能力, 在积分区域  $\Omega$  上随机产生测试集  $\Phi = \{(x_1, Y_1^*), (x_2, Y_2^*), \dots, (x_n, Y_n^*)\}$ , 其中  $Y_i^*$  为式(2)的真解在  $x_i$  处的取值。定义积分方程的真解  $Y^*$  与神经网络学习得到的近似解  $F(x, \theta)$  之间的泛化误差为

$$E_{\text{accuracy}} = \frac{1}{n} \sum_{j=1}^n (Y_i^* - F(x_i, \theta))^2 \quad (7)$$

综上所述, 残差神经网络求解 Fredholm 积分方程的算法如下:

- 初始化  $\theta$ , 即给所有权重和阈值随机赋值。
- 随机产生一组训练样本  $\{x_1, x_2, \dots, x_n\}$ 。
- 前向传播得到  $F(x_i, \theta)$ ,  $i = 1, 2, \dots, n$ 。
- 反向传播, 更新  $\theta$ , 在此采用了 Adma 优化器。
- 判断迭代次数是否大于给定的最大迭代次数, 如成立, 转步骤 f), 否则转步骤 b)。
- 输出整个神经网络参数值  $\theta$ , 得到残差神经网络的近似解。

残差神经网络模型求解 Fredholm 积分方程时, 极小化问题(6)中参数  $\theta$  的维数只与神经网络结构有关, 计算式(5)时乘法、除法的次数为  $n(n+2)$ , 只与训练集的数据个数  $n$  有关, 只有函数  $k(x, y)$  与  $g(x)$  的计算与方

程的维数有关。文献[23-24]给出残差神经网络模型的后验误差与蒙特卡罗方法的理论误差  $O(1/\sqrt{n})$  同阶, 与问题的维数无关。因此残差神经网络模型求解 Fredholm 积分方程时, 计算量和误差依赖于神经网络的结构和训练集的数据个数, 而不依赖于问题的维数, 因此在求解高维 Fredholm 积分方程时训练残差神经网络时计算量不会显著增长。而传统的数值方法的误差和计算量都依赖于问题的维数, 计算量随着维数的增加指数增长, 很难真正求解高维问题, 如文献[20, 25]中的方法在求解二维或四维 Fredholm 积分方程时精度很高, 但很难推广到高维问题。

本文采用的框架是 TensorFlow, 它是一个开源的、基于 Python 的机器学习框架, 由 Google 公司开发, 在图像处理和分类、物体检测、机器翻译、推荐系统和自然语言处理等领域有着广泛的应用, 是目前最热门的机器学习框架。本文构建的残差神经网络深度为 6 层(如图 3 所示), 含有两个残差块。训练网络过程中使用 Adam 优化器, 它是一阶优化算法, 可以代替传统的随机梯度下降法, 迭代更新神经网络参数  $\theta$ 。本文所用的神经网络的 Tensorboard 可视化结构图如图 4 所示, 其中 random\_unif 表示神经网络的输入层, fc5 表示神经网络的输出层, fc1、fc2、fc3、fc4 表示神经网络的隐藏层, loss 表示计算损失函数的模块, accuracy 表示计算泛化误差的模块, Adam\_optimi... 表示优化算法模块, 箭头表示数据流动方向。比较图 3 和图 4 可知, 数据流在网络训练过程中与网络设计保持一致。

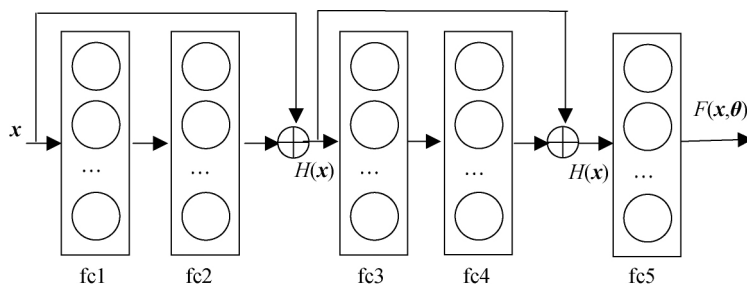


图 3 残差神经网络示意图

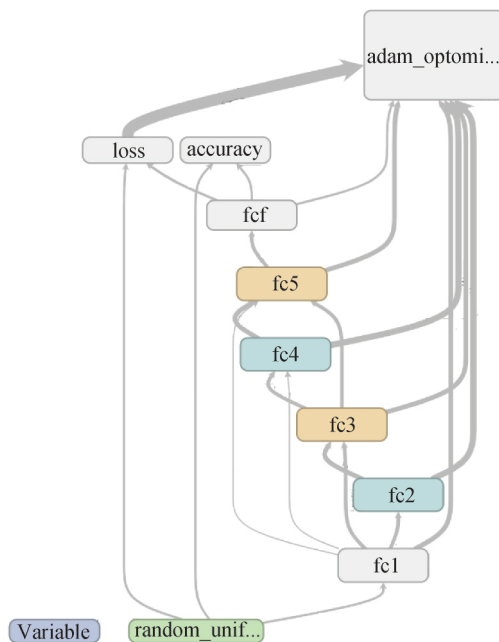


图 4 Tensorboard 神经网络可视化结构图

### 3 数值实验分析

为了说明残差神经网络方法对 Fredholm 积分方程数值求解的有效性, 本文对多个 Fredholm 积分方程进行了数值求解。在计算过程中, 在积分区域  $\Omega$  内随机取  $n=1000$  个点作为训练集训练残差神经网络, 网络结构如图 3 所示, 训练迭代次数为 2000 步。为了说明残差神经网络的泛化能力, 每次训练完成后, 在积分区

域  $\Omega$  内随机取  $n = 1000$  个点作为测试集, 计算由式(7)定义的泛化误差。

例 1 求解下述第二类 Fredholm 积分方程<sup>[17]</sup>:

$$f(x) - \frac{3}{2} \int_0^1 e^{|x-y|} f(y) dy + 1 = 0 \quad (8)$$

该方程的精确解是

$$f^*(x) = \frac{3}{4} \left[ \frac{3 + e^2}{e^4 - 9} (e^{2x} + e^{2(1-x)}) - 1 \right] \quad (9)$$

Fredholm 积分方程(8)中核函数是  $k(x, y) = e^{|x-y|}$ , 自由项是  $g(x) = -1$ , 代入式(5)得残差神经网络的损失函数。

图 5 和图 6 分别给出例 1 网络训练过程中损失函数和泛化误差的收敛曲线, 表 1 给出损失函数和泛化误差迭代结果的部分关键数据。

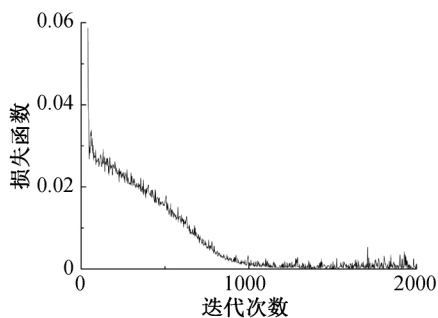


图 5 例 1 的损失函数收敛曲线

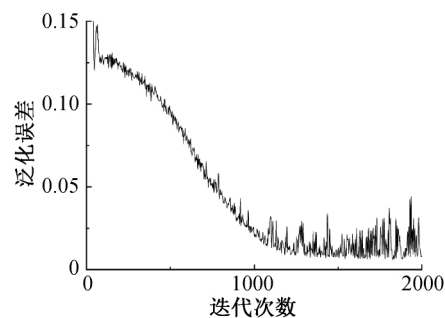


图 6 例 1 的泛化误差收敛曲线

表 1 例 1 的损失函数值 (loss 函数值) 和泛化误差 ( $E_{accuracy}$ ) 部分迭代结果

迭代次数	$E_{accuracy}$	loss 函数值	迭代次数	$E_{accuracy}$	loss 函数值
5	0.63121	0.58428	611	0.07362	0.01016
6	0.62081	0.55762	...	...	...
...	...	...	1008	0.02080	0.00086
206	0.12333	0.02444	1009	0.01955	0.00081
207	0.12324	0.02339	...	...	...
...	...	...	1999	0.00778	0.00099
610	0.07866	0.01187	2000	0.00820	0.00028

图 5 给出的损失函数收敛曲线表示式(5)定义的损失函数在训练迭代过程中的变化。在训练过程中, 前 200 次迭代损失函数下降很快, 1000 次迭代后损失函数下降到稳定状态, 此后损失函数与  $10^{-3}$  同阶。图 6 给出由式(7)定义的泛化误差收敛曲线的变化过程。迭代前 200 步泛化误差迅速下降, 以一定速度下降到 1000 步后, 泛化误差稳定下来, 与  $10^{-2}$  同阶。由此可见泛化误差和损失函数的收敛基本同步。

由图 5、图 6 和表 1 可知, 残差神经网络求解 Fredholm 积分方程有很好的收敛性, 从而说明该方法是可行的和有效的。

例 2 考虑三维的 Fredholm 积分方程:

$$f(x, y, z) - \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 k(x, y, z, s, t, v) f(s, t, v) ds dt dv = g(x, y, z), x, y, z \in [-1, 1],$$

其中:  $k(x, y, z, s, t, v) = e^{-xs-yt-zv}$ ,  $g(x, y, z) = 1 - \frac{1}{xyz} (e^{-x} - e^x)(e^{-y} - e^y)(e^{-z} - e^z)$ , 方程的精确解是  $f^*(x, y, z) = 1$ 。

图 7 和图 8 分别给出例 2 网络训练过程中损失函数和泛化误差的收敛曲线, 表 2 给出损失函数和泛化误差迭代结果的部分数据。网络训练迭代 300 步时, 损失函数和泛化误差同时收敛到稳定状态, 损失函数与  $10^{-3}$  同阶, 泛化误差与  $10^{-2}$  同阶。因此, 残差神经网络求解 Fredholm 积分方程有很好的收敛性, 表明该方法是可行的和有效的。

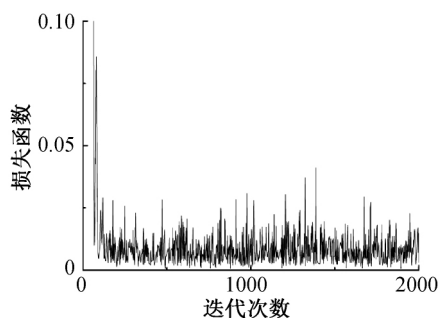


图 7 例 2 的损失函数收敛曲线

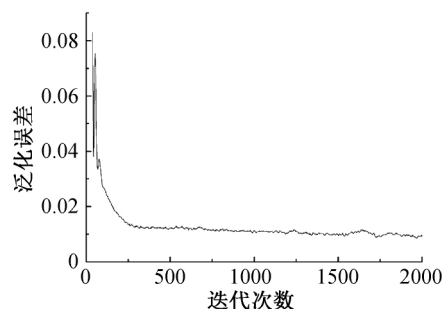


图 8 例 2 的泛化误差收敛曲线

表 2 例 2 的损失函数值 (loss 函数值) 和泛化误差 (E<sub>accuracy</sub>) 部分迭代结果

迭代次数	E <sub>accuracy</sub>	loss 函数值	迭代次数	E <sub>accuracy</sub>	loss 函数值
5	0.69700	53.410	303	0.01279	0.00919
6	0.68082	50.947	...	...	...
...	...	...	1491	0.01020	0.01269
141	0.02179	0.01305	1492	0.01008	0.00233
142	0.02165	0.01468	...	...	...
...	...	...	1999	0.00949	0.00703
302	0.01314	0.00403	2000	0.00927	0.00306

例 3 考虑四维 Fredholm 积分方程<sup>[25]</sup>:

$$f(x, y, z, w) = \frac{15}{16}xyzw - xyzw \int_0^1 \int_0^1 \int_0^1 \int_0^1 f(s, t, v, r) ds dt dv dr, \quad (x, y, z, w) \in [0, 1]^4,$$

其中:  $k(x, y, z, w, s, t, v, r) = xyzw, g(x, y, z, w) = \frac{15}{16}xyzw$ 。其精确解为  $f^*(x, y, z, w) = xyzw$ 。

图 9 和图 10 分别给出例 3 网络训练过程中损失函数和泛化误差的收敛曲线, 表 3 给出损失函数和泛化误差迭代结果的部分数据。网络训练迭代 200 步时, 损失函数和泛化误差同时收敛到稳定状态, 损失函数与  $10^{-3}$  同阶, 泛化误差与  $10^{-2}$  同阶。

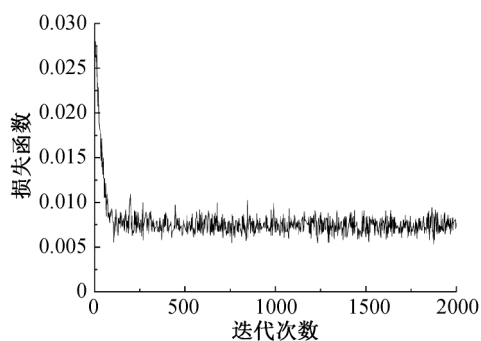


图 9 例 3 的损失函数收敛曲线

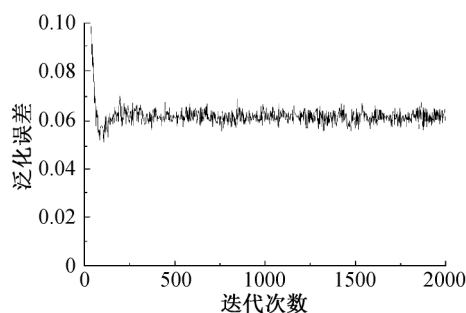


图 10 例 3 的泛化误差收敛曲线

表 3 例 3 的损失函数值 (loss 函数值) 和泛化误差 (E<sub>accuracy</sub>) 部分迭代结果

迭代次数	E <sub>accuracy</sub>	loss 函数值	迭代次数	E <sub>accuracy</sub>	loss 函数值
5	0.14786	0.02796	208	0.06205	0.00626
6	0.14415	0.02203	...	...	...
...	...	...	1541	0.06354	0.00778
105	0.05410	0.00776	1542	0.06310	0.00741
106	0.05047	0.00553	...	...	...
...	...	...	1999	0.06163	0.00767
207	0.06556	0.00851	2000	0.06157	0.00727

数值实验表明,残差神经网络模型求解不同维数的 Fredholm 积分方程时,当训练集取的数据个数  $n$  相同时,损失函数和泛化误差都能取得很好的收敛性,说明本文提出的方法可以有效求解高维 Fredholm 积分方程,而计算量不会显著增长。数值结果显示损失函数和泛化误差的误差阶有时优于蒙特卡罗方法的理论误差  $O(1/\sqrt{n})$ 。

#### 4 结 论

本文基于深度神经网络可以有效解决高维问题的特性,构造残差神经网络求解 Fredholm 积分方程,此方法不仅可以解决一维问题,对二维以及更高维度的积分方程也同样有效,且计算量没有显著提高,因此在求解高维 Fredholm 积分方程有显著优势。该方法也可以推广到求解其它类型的积分方程。但该算法计算精度还不够高,可以通过优化网络结构进一步提高计算精度,提高该方法的实用性,扩大使用范围。

#### 参考文献:

- [1] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The Bulletin of Mathematical Biophysics, 1943, 5(4): 115-133.
- [2] Hinton G E. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.
- [3] Sermanet P, Chintala S, Lecun Y. Convolutional neural networks applied to house numbers digit classification[C]//Proceedings of the 21st International Conference on Pattern Recognition. Tsukuba: IEEE, 2012: 3288-3291.
- [4] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB/OL]. (2020-01-07)[2016-05-19]. <https://arXiv:1409.0473>.
- [5] Huang P, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. New York: Association for Computing Machinery, 2013: 2333-2338.
- [6] Wang Y Q, Huang M L, Zhu X Y, et al. Attention-based LSTM for aspect-level sentiment classification[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin: Association for Computational Linguistics, 2016: 606-615.
- [7] Li X, Bing L, Lam W, et al. Transformation networks for target-oriented sentiment classification[EB/OL]. (2020-01-07)[2018-05-03]. <https://arXiv.org/abs/1805.01086>.
- [8] Yi C, Chen Y, Lu Z. Improved gradient-based neural networks for online solution of Lyapunov matrix equation[J]. Information Processing Letters, 2011, 111(16): 780-786.
- [9] E W, 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.
- [10] E W N, Han J Q, Jentzen A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations[J]. Communications in Mathematics and Statistics, 2017, 5(4): 349-380.
- [11] Han J Q, Jentzen A, E W N. Solving high-dimensional partial differential equations using deep learning[J]. Proceedings of the National Academy of Sciences of the United States of America, 2018, 115(34): 8505-8510.
- [12] Ray D, Hesthaven J S. An artificial neural network as a troubled-cell indicator[J]. Journal of Computational Physics, 2018, 367: 166-191.
- [13] 吕涛, 黄晋. 积分方程的高精度算法[M]. 北京: 科学出版社, 2013: 189-190.
- [14] Ren Y H, Zhang B, Qiao H. A simple Taylor-series expansion method for a class of second kind integral equations[J]. Journal of Computational and Applied Mathematics, 1999, 110(1): 15-24.
- [15] Babolian E, Shamsavaran A. Numerical solution of nonlinear Fredholm integral equations of the second kind using Haar wavelets[J]. Journal of Computational and Applied Mathematics, 2009, 225(1): 87-95.
- [16] Xiao J Y, Wen L H, Zhang D. Solving second kind Fredholm integral equations by periodic wavelet Galerkin method[J]. Applied Mathematics and Computation, 2006, 175(1): 508-518.
- [17] Jin C M, Ding J. Solving Fredholm integral equations via a piecewise linear maximum entropy method[J]. Journal of Computational and Applied Mathematics, 2016, 304: 130-137.

- [18] Han G, Wang J. Extrapolation of Nyström solution for two-dimensional nonlinear Fredholm integral equations[J]. Journal of Computational and Applied Mathematics, 2001, 134(1/2): 259-268.
- [19] Borzabadi A H, Fard O S. A numerical scheme for a class of nonlinear Fredholm integral equations of the second kind[J]. Journal of Computational and Applied Mathematics, 2009, 232(2): 449-454.
- [20] Kazemi M, Golshan H M, Ezzati R, et al. New approach to solve two-dimensional Fredholm integral equations[J]. Journal of Computational and Applied Mathematics, 2019, 354: 66-79.
- [21] 伊恩·古德费洛, 约书亚·本吉奥, 亚伦·库维尔. 深度学习[M]. 赵申剑, 黎戎君, 符天凡, 等译. 北京: 邮电出版社, 2016: 105-140.
- [22] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2016: 770-778.
- [23] E W, Ma C, Wang Q. A priori estimates of the population risk for residual networks[EB/OL]. (2020-01-07) [2019-05-31]. <https://arxiv.org/abs/1903.02154>.
- [24] E W, Ma C, Wu L. Barron spaces and the compositional function spaces for neural network models[EB/OL]. (2020-01-07) [2019-06-18]. <https://arxiv.org/abs/1906.08039>.
- [25] Liu H Y, Huang J, Pan Y B, et al. Barycentric interpolation collocation methods for solving linear and nonlinear high-dimensional Fredholm integral equations[J]. Journal of Computational and Applied Mathematics, 2018, 327: 141-154.

(责任编辑: 康 锋)