

用前馈神经网络解稳态对流扩散方程

孟柳君, 成蓉华

云南财经大学, 云南 昆明

收稿日期: 2021年12月7日; 录用日期: 2021年12月28日; 发布日期: 2022年1月10日

摘 要

本文主要用前馈神经网络求解一维稳态对流扩散方程, 因方程中含有一个小参数, 所以用传统方法不容易达到理想效果, 本文通过构造神经网络模拟方程并与其精确解作对比, 选取 $\varepsilon = \{0.1, 0.01\}$ 借助软件进行模拟, 计算其误差, 做出图像。

关键词

前馈神经网络, 偏微分方程

Feedforward Neural Networks for Solving Steady-State Convection Diffusion Equation

Liujun Meng, Ronghuan Cheng

Yunnan University of Finance and Economics, Kunming Yunnan

Received: Dec. 7th, 2021; accepted: Dec. 28th, 2021; published: Jan. 10th, 2022

Abstract

This paper mainly uses feedforward neural network to solve one-dimensional steady-state convection diffusion equation. The equation contains a small parameter, so the traditional method cannot achieve the ideal effect. This paper constructs the neural network simulation equation and then makes comparison with its accurate solution. This paper chooses parameter $\varepsilon = \{0.1, 0.01\}$, uses computer software to simulate, calculates its error, and makes an image.

Keywords

Feedforward Neural Networks, Partial Differential Equation

Copyright © 2022 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

在物理和工程的应用过程中很多问题最后都可以转化成一个求解偏微分方程的问题,但在物理中,很多的偏微分方程的精确解是很难得到的,这样以来数值方法求解偏微分方程就很有必要了。传统的求解偏微分方程的数值方法有很多,如欧拉法、龙格库塔法、有限差分法、有限元法、有限体积法以及谱方法。近几年随着计算机和人工智能的快速发展,用神经网络解偏微分方程成了新的求解微分方程的数值技术。人工神经网络是基于大脑中神经网络的工作方式,人为构建的一个网络,可以对大数据信息进行识别、分类、学习、运算处理,从而解决一些简单计算机无法处理的复杂问题,并且能极大地提升一些复杂问题的处理速度。

1989年Cybenko证明了具有单层神经网络可任意精度逼近一个函数[1],为神经网络逼近微分方程打下了理论基础。近几年国内外对神经网络解微分方程都有较高的关注,Raissi等人设计了一个包含物理信息的神经网络来求解一些局部非线性偏微分方程[2]。在国内2017年鄂维南院士[3]讨论了使用连续动力系统建模高维非线性函数的想法,即微分方程与深度学习之间的关系,更是掀起了国内讨论的热潮。

本文借助前馈神经网络模型模拟对流扩散方程在实际的科学研究和工程技术中产生的一些问题。

2. 对流稳态方程介绍

对流扩散方程被广泛地应用于许多自然现象的表达之中,例如水和大气中的污染物质浓度的扩散、海水盐度、流体流动与传热、电化学反应等,还应用于环境科学、能源开发、电子科技等[4]。

n 维稳态对流扩散方程形式如下[5]:

$$-\varepsilon \Delta u + \lambda \nabla u = f, \quad u = u(x), \quad x \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}^n$$

其中 λ 是给定的常量,本文我们讨论一维稳态对流方程问题,也就是 $n=1$ 的情况,我们可得如下方程

$$-\varepsilon \frac{\partial^2 u}{\partial x^2} + \lambda \frac{\partial u}{\partial x} = f(x) \quad (2)$$

对于实际问题的计算需加上边界条件才能模拟,故为上面方程加上狄里克莱边界条件,于是对于方程(2)可化为如下形式。

$$\begin{cases} -\varepsilon \frac{\partial^2 u}{\partial x^2} + \lambda \frac{\partial u}{\partial x} = f(x) \\ u(0) = 0, u(1) = 0 \end{cases} \quad (3)$$

其中 ε 为大于零的微小常数, λ 为已知常数,在 f 为恒定常数,问题(2)有如下解析解。

$$u(x) = \frac{f}{\lambda} \left(x - \frac{\frac{\lambda x}{\varepsilon} - 1}{e^{\frac{\lambda}{\varepsilon}} - 1} \right)$$

本文将对方程(3)在定点条件 $\lambda=1$, $f=1$ 的条件下进行数值模拟, 并与精确解作对比。

3. 神经网络介绍

人工神经网络的结构类似人脑神经网络的构成方式, 是由一个个神经元(节点)组成的。人工神经网络的工作原理也类似大脑对信息的处理方式: 信息输入到输入层, 经过一个个相连的神经元(节点)接收、处理信息, 最终传递到输出层。

神经网络是由很多神经元相互连接而形成的网络。它是通过对训练样本的反复学习, 并通过每次学习对连接神经元的权值和阈值进行修正, 最终使训练的网络达到能够完全识别训练样本, 即完成对一件事物的认识并记忆。

不论什么样的神经网络模型, 其中一个最小的信息处理单元就是神经元, 是神经网络的基本处理单元。图 1 给出了传统人工神经元结构模型图。

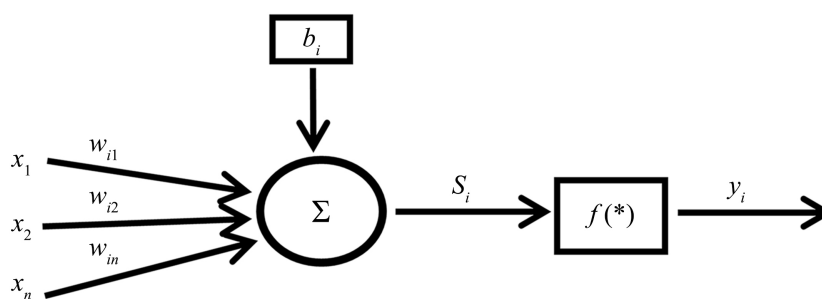


Figure 1. Schematic diagram of neurons

图 1. 神经元示意图

以数学式表达出来:

$$S_i = \sum_{j=1}^n (w_{ij} x_j) + b_i$$

$$y_i = f(S_i)$$

其中 $x_1, \dots, x_n$ 为输入数据 $w_{11}, \dots, w_{in}$ 为权值, $b_1, \dots, b_n$ 为偏值, S_i 为第 i 个神经元的输入, $f(*)$ 为激活函数, y_i 为第 i 个神经元的输出。

多个神经元连接组成神经网络, 前馈神经网络又称为多层感知器, 是一种最简单的网络结构形式, 本文主要前馈神经网络解微分方程, 图 2 展示了含有一层隐藏层的前馈神经网络结构。

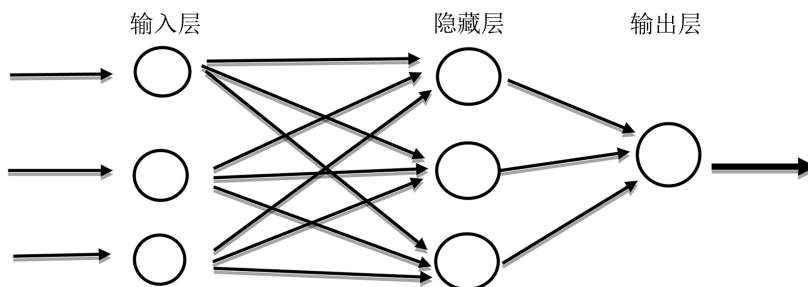


Figure 2. Neural network diagram

图 2. 神经网络示意图

4. 神经网络解微分方程的步骤

对于一个给定边界条件的微分方程

$$\begin{cases} \Re u(x) = f(x), & x \in \Omega \\ u(x) = g(x), & x \in \partial\Omega \end{cases}$$

其中 $u(x)$ 是一个未知函数, $\Re$ 为已知微分算子, $f(x)$ 和 $g(x)$ 为已知函数。神经网络解微分方程的步骤如下:

1) 在定义域 Ω 中选取内部节点 $x_i (i=1,2,\dots,N)$, 在 $\partial\Omega$ 上取好边界点 $x_j^b (j=1,2,\dots,M)$ 其中 M, N 都为正整数。

2) 构造网络函数, 假定 $u(x) \approx u(x; \theta) \equiv F^{W,b}(x)$, 其中 $\theta = (W, b)$ 为网络中未知参数。

3) 构造损失函数, $L(\theta) = \frac{1}{N} \sum_{i=1}^N (\Re(u(x; \theta)) - f(x))^2 \Big|_{x=x_i} + \frac{1}{M} \sum_{j=1}^M (u(x; \theta) - g(x))^2 \Big|_{x=x_j^b}$ 。

4) 解关于 3) 中损失函数的最小值, 求出网络参数 W^*, b^* ,

$$(W^*, b^*) = \arg(\min(L(\theta)))$$

5) 得到未知函数的逼近函数 $u(x, \theta^*) \approx F^{W^*, b^*}(x)$ 。

5. 数值模拟

本文选取 $\varepsilon = \{0.1, 0.01\}$ 三种情况进行模拟, 借助软件 Python 中的 TensorFlow 框架进行计算。在神经网络训练过程中选取含有五层隐藏层的前馈神经网络进行计算, 训练 20,000 次, 画出数值解和精确解的图像, 并计算出误差。图 3 是 $\varepsilon = 0.1$ 时方程(3)的数值解与精确解的图像。图 4 是 $\varepsilon = 0.01$ 时方程(3)的数值解与精确解的图像。

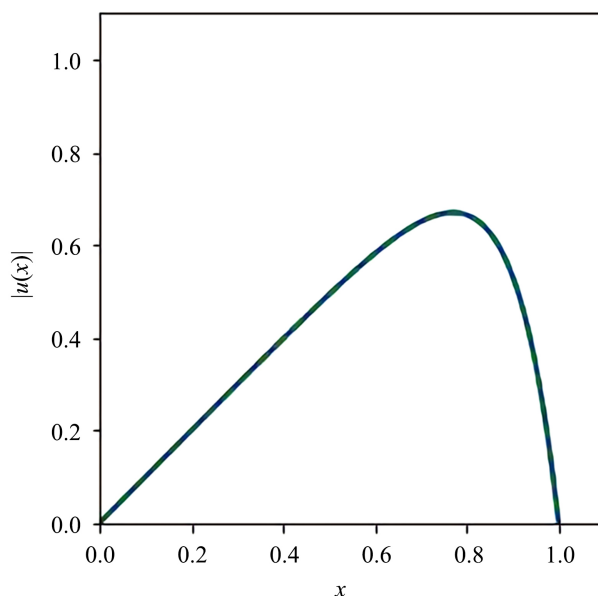


Figure 3. Images of exact solutions (blue line) and approximate solutions (green line); error is $5.69\text{e-}05$

图 3. 精确解(蓝线)与近似解(绿线)的图像; 误差为 $5.69\text{e-}05$

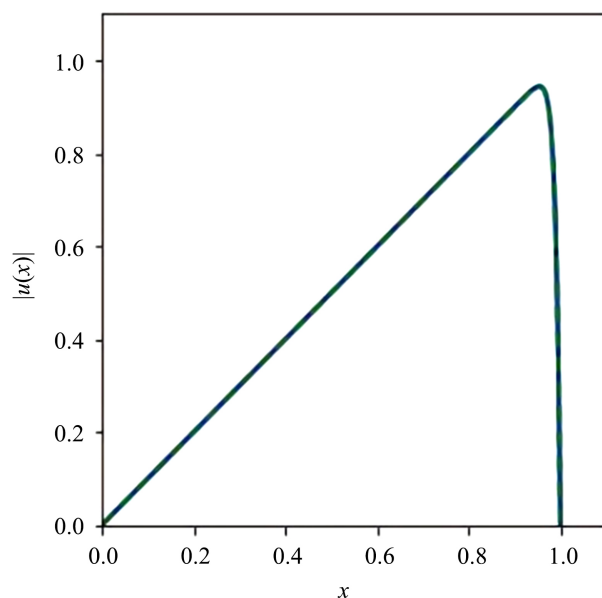


Figure 4. Images of exact solutions (blue line) and approximate solutions (green line); error is $9.01\text{e-}05$

图 4. 精确解(蓝线)与近似解(绿线)的图像; 误差为 $9.01\text{e-}05$

6. 结论

本文给出了前馈神经网络模型, 及用前馈神经网络解偏微分方程的详细步骤。采用前馈神经网络解了一维稳态对流方程, 展示了神经网络逼近函数的误差, 观察发现对于含有小参数的微分方程, 神经网络也有不错的表现。

参考文献

- [1] Cybenko, G. (1989) Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, **2**, 303-314. <https://doi.org/10.1007/BF02551274>
- [2] Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019) Physics Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational Physics*, **378**, 686-707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [3] Weinan, E. (2017) A Proposal on Machine Learning via Dynamical Systems. *Communications in Mathematics and Statistics*, **5**, 1-11. <https://doi.org/10.1007/s40304-017-0103-z>
- [4] 毕超. 计算流体力学有限元方法及其编程详解[M]. 北京: 机械工业出版社, 2013.
- [5] Brezzi, F. and Russo, A. (1994) Choosing Bubbles for Advection-Diffusion Problems. *Mathematical Models and Methods in Applied Sciences*, **4**, 571. <https://doi.org/10.1142/S0218202594000327>