Advances in Applied Mathematics 应用数学进展, 2022, 11(4), 1870-1875
Published Online April 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.114204

Hans汉斯

三阶常微分方程的神经网络模型分析

伍　阳，杨云磊*

贵州大学，贵州　贵阳

收稿日期：2022年3月20日；录用日期：2022年4月14日；发布日期：2022年4月22日

摘　要

本文研究神经网络模型对一类三阶常微分方程数值解的影响。首先，分析网络结构在神经网络模型求解常微分方程中的重要性，接着，探究单隐层前馈神经网络的网络隐层激活函数，选择几类正交多项式来消除隐层，构造不同类型的神经网络模型，利用极限习机(ELM)算法求解网络权值，最后利用数值实验模拟展示不同的神经网络模型形成的影响。

关键词

ODE，神经网络模型，正交多项式

Comparison of Several Neural Network Methods for a Class of Third Order Ordinary Differential Equations

Yang Wu, Yunlei Yang*

Guizhou University, Guiyang Guizhou

Received: Mar. 20th, 2022; accepted: Apr. 14th, 2022; published: Apr. 22nd, 2022

Abstract

In this paper, the influence of neural network model on the numerical solution of a class of third-order ordinary differential equations is studied. First, the importance of network structure in solving ordinary differential equations by neural network model is analyzed. Then, the network hidden layer activation function of single hidden layer feed forward neural network is explored. Several orthogonal polynomials are selected to eliminate hiding, and different types of neural

*通讯作者。

文章引用: 伍阳, 杨云磊. 三阶常微分方程的神经网络模型分析[J]. 应用数学进展, 2022, 11(4): 1870-1875.
DOI: 10.12677/aam.2022.114204

network models are constructed. The limit learning machine (ELM) algorithm is used to solve the network weights. Finally, numerical experiments are used to simulate the influence of different neural network models.

Keywords

ODE, Neural Network Model, Orthogonal Polynomial

1. 引言

科学和工程中物理现象的数学模型往往可以表示成以下形式的初值问题：

$$y''' = f\left(x, y, y', y''\right)$$
$$y(a) = y_0, y'(a) = y_1, y''(a) = y_2$$

要得到(1)的解析解往往非常困难或者无法求解，所以有必要研究该问题的数值方法。长期以来，人们提出了许多求解常微分方程的数值方法，Mohammed [1]提供了一种三阶常微分方程解的三隐式混合线性多步方法，Olabode [2]提出了一种三阶常微分方程的分块多步法，Agboola [3]提出了一种微分变换方法解三阶常微分方程，Hashim [4]利用伯恩斯坦多项式方法的运算矩阵直接求解三阶方程，并应用于流体流动方程。求解三阶常微分方程问题的数值方法有许多的研究结果，有的计算精度较高，但存在着随着样本量的增加，执行时间迅速增加的问题。

用正交多项式消除隐层已经在神经网络的微分方程数值解法中得到了广泛的应用，Lu [5]提出了用勒让德神经网络算法求解某些风险模型的破产概率，Ma 等人[6]提出一种改进的求解连续时间一维资产定价模型的价格红利函数的三角神经网络算法，陈英皞[7]用拉盖尔神经网络数值求解广义 Black-Scholes 微分方程。

本文借用神经网络模型对求解三阶常微分方程模型选择问题进行一些探究。

2. 单隐层前馈神经网络

神经网络中最基本的单元是神经元模型，细胞体分为两部分，前一部分计算总输入值，即输入信号的加权和，后一部分计算总输入值与该神经元阈值的插值，然后通过激活函数处理，产生输出从轴突传送给其他神经元。

人工神经网络模仿了生物神经网络。将神经元模型按层连接，就能得到单层前馈神经网络，其中，我们将隐藏层的基函数选择为正交多项式，就能得到如图 1 所示的网络结构。单隐层前馈神经网络由输入层、隐含层、输出层组成，其中 x_i 是我们的输入，b 表偏置，φ_i 是我们所选用的正交多项式，y_i 是输出。可简单模拟生物神经网络，每层神经元与下一层神经元连接，神经元之间不存在跨层连接、同层连接，输入层用于数据的输入，隐含层与输出层神经元对数据进行加工。

3. 神经网络对一般三阶微分方程求解步骤

对于给定的任意一般三阶常微分方程：

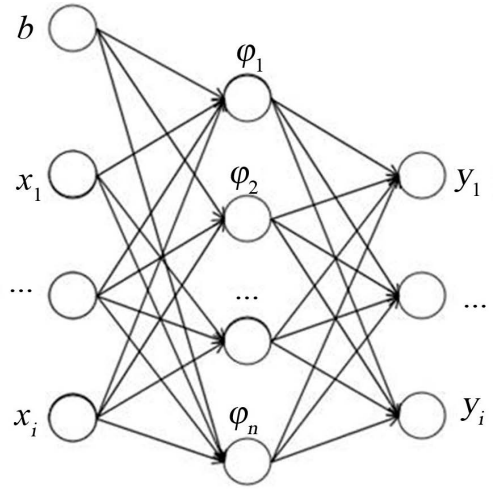

Figure 1. The neural network diagram
图 1. 神经网络示意图

$$\begin{cases} y''' + ay'' + by' + cy + d = f(x) \\ y(e) = y_0, y'(k) = y_1, y''(k) = y_2 \end{cases}$$

使用正交多项式作为隐藏层基函数代替近似解，使用 ELM 算法计算权值，步骤如下：

1) 构造方程的近似解 $\hat{y}(x) = \sum_{i=0}^{N} \beta_i \varphi_i(x)$，并把近似解带入方程组。

$$\sum_{i=0}^{N} \beta_i \varphi_i'''(x) + a\sum_{i=0}^{N} \beta_i \varphi_i''(x) + b\sum_{i=0}^{N} \beta_i \varphi_i'(x) + c\sum_{i=0}^{N} \beta_i \varphi_i(x) = f(x) - d$$

$$\sum_{i=0}^{N} \beta_i \varphi_i(e) = y_0, \sum_{i=0}^{N} \beta_i \varphi_i'(k) = y_1, \sum_{i=0}^{N} \beta_i \varphi_i''(g) = y_2$$

2) 剖分定义域 Ω，选取内部节点 $x_i (i = 1,2,\cdots N)$，将节点带入方程：

$$\sum_{i=0}^{N} \left(\varphi_i'''(x_j) + a\varphi_i''(x_j) + b\varphi_i'(x_j) + c\varphi_i(x_j) \right) \beta_i = f(x_j) - d, j = 0,1,2,\cdots, M$$

3) 形成数值矩阵 $H\beta = T$。

4) 使用 ELM 算法求解权值矩阵 $\beta = H^{+}F$，其中 H^{+} 为矩阵 H 的 Moore-Penrose 广义逆。

4. 数值实验

在解域内均匀选择测试点，并在这些测试点上比较真实解与近似解的插值，才用均方误差(MSE)来观察基函数的选择对误差的影响。MSE 的定义如下：

$$\text{MSE} = \frac{1}{S}\sum_{i=1}^{S} \left[y(x_i) - y(\hat{x}_i) \right]^2$$

例 1 考虑方程：

$$\begin{cases} y''' = e^x \\ y(0) = 3, y'(0) = 1, y''(0) = 5, \end{cases}$$

方程的精确解为：$y_{\text{exact}}(x)=2+2x+\mathrm{e}^{x}$。

我们选取神经元个数 $N=10$，当选择测试点作为区间[0, 1]中的点时，精确解、Legendre 神经网络、Chebyshev 神经网络和 Hermite 神经网络结果的比较记录在表 1 中 。

Table 1. Numerical results using different neural network models
表 1. 采用不同神经网络模型的数值结果

x	精确解	LNN 解	LNN 误差	CNN 解	CNN 误差	HNN 解	HNN 误差
0	3.000000000000	3.000000000000	0	3.000000000000	0	2.999999999998	1.1e−12
0.1	3.125170918076	3.125170918068	6.7e−12	3.125170918068	6.9e−12	3.125170918068	6.7e−12
0.2	3.301402758160	3.301402758106	5.3e−11	3.301402758106	5.3e−11	3.301402758107	5.2e−11
0.3	3.529858807576	3.529858807443	1.3e−10	3.529858807443	1.3e−10	3.529858807446	1.2e−10
0.4	3.811824697641	3.811824697415	2.2e−10	3.811824697415	2.2e−10	3.811824697419	2.2e−10
0.5	4.148721270700	4.148721270350	3.4e−10	4.148721270349	3.5e−10	4.148721270356	3.4e−10
0.6	4.542118800390	4.542118799868	5.2e−10	4.542118799867	5.2e−10	4.542118799875	5.1e−10
0.7	4.993752707470	4.993752706743	7.2e−10	4.993752706741	7.2e−10	4.993752706751	7.1e−10
0.8	5.505540928492	5.505540927546	9.4e−10	5.505540927545	9.4e−10	5.505540927556	9.3e−10
0.9	6.079603111156	6.079603109967	1.1e−9	6.079603109959	1.1e−9	6.079603109972	1.1e−9
1.0	6.718281828459	6.718281826972	1.4e−9	6.718281826970	1.4e−9	6.718281826984	1.4e−9

由表 1 可以看出，Legendre 神经网络算法得到的误差精度为 o(10-8)，MSE 约为 5.029218803638438e−19。Chebyshev 神经网络算法得到的误差精度为 o(10-8)，MSE 约为 5.042613965704128e−19。Hermite 神经网络算法得到的误差精度为 o(10-8)，MSE 约为 4.928755732194928e−19。由图 2 可以看出，三种神经网络模型的性能是相似的，误差曲线图基本上重合。

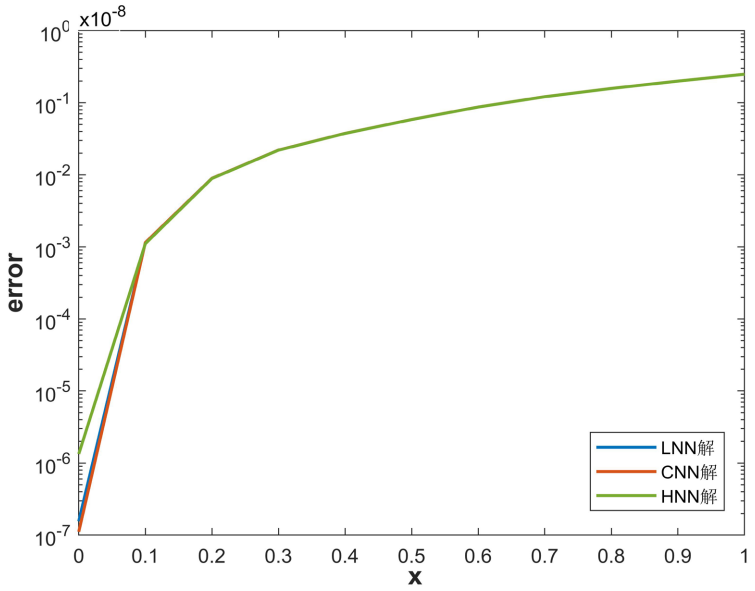

Figure 2. Error curves of comparative example 1 with different neural network models
图 2. 不同神经网络模型对例 1 误差曲线图

例 2 考虑方程：

$$\begin{cases} y''' = -y \\ y(0)=1, y'(0)=-1, y''(0)=1, \end{cases}$$

方程的精确解为： $y_{\text{exact}}(x)=\mathrm{e}^{-x}$。

我们选取神经元个数 $N=10$，当选择测试点作为区间[0, 1]中的点时，精确解、Legendre 神经网络、Chebyshev 神经网络和 Hermite 神经网络结果的比较记录在表 2 中。

Table 2. Numerical results using different neural network models
表 2. 例 2 采用不同神经网络模型的数值结果

x	精确解	LNN 解	LNN 误差	CNN 解	CNN 误差	HNN 解	HNN 误差
0	1	0.999999999999	1.0e−13	1.000000000000	0	1.000000000021	2.1e−11
0.1	0.904837418035	0.904837418032	3.0e−12	0.904837418033	2.2e−12	0.904837418043	7.9e−12
0.2	0.818730753077	0.818730753055	2.2e−11	0.818730753057	2.0e−11	0.818730753056	2.1e−11
0.3	0.740818220681	0.740818220628	5.3e−11	0.740818220629	5.1e−11	0.740818220618	6.3e−11
0.4	0.670320046035	0.670320045945	9.0e−11	0.670320045946	9.0e−11	0.670320045924	1.1e−10
0.5	0.606530659712	0.606530659572	1.4e−10	0.606530659573	1.4e−10	0.606530659542	1.7e−10
0.6	0.548811636094	0.548811635885	2.0e−10	0.548811635886	2.0e−10	0.548811635845	2.4e−10
0.7	0.496585303791	0.496585303502	2.8e−10	0.496585303791	2.8e−10	0.496585303452	3.3e−10
0.8	0.449328964117	0.449328963742	3.7e−10	0.449328964117	3.7e−10	0.449328963683	4.3e−10
0.9	0.406569659740	0.406569659268	4.7e−10	0.406569659740	4.7e−10	0.406569659200	5.3e−10
1.0	0.367879441171	0.367879440587	5.8e−10	0.367879441171	5.8e−10	0.367879440511	6.5e−10

由表 2 可以看出，Legendre 神经网络算法得到的误差精度为 o(10-10)，MSE 约为 5.029218803638438e−19。Chebyshev 神经网络算法得到的误差精度为 o(10-10)，MSE 约为 5.042613965704128e−19。Hermite 神经网络算法得到的误差精度为 o(10-10)，MSE 约为 4.928755732194928e−19。由图 3 可以看出，三种神经网络模型的性能是相似的，误差曲线图基本上重合。

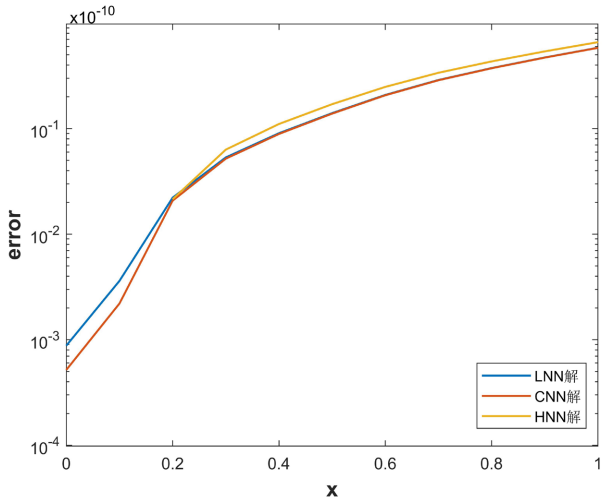

Figure 3. Error curves of comparative example 2 with different neural network models
图 3. 不同神经网络模型对例 2 误差曲线图

5. 结论

本文通过建立不同神经网络模型，对求解一类三阶常微分方程数值解的误差进行探究，通过神经网络方法获取的近似解与数值解之间的误差，探究出选择 Legendre 多项式、Chebyshev 多项式、Hermite 多项式作为神经网络的基函数对数值解的误差量级影响较小，为以后用神经网络解三阶常微分方程基函数的选择上，提供了一些参考。

项目基金

贵州省科技计划项目(No. QKHJC-ZK[2021]YB017)；贵州大学引进人才项目(No. GzuRJHZ[2019]047)；黔科合平台人才[2020]5016。

参考文献

[1] Mohammed, U. and Adeniyi, R.B. (2014) A Three Step Implicit Hybrid Linear Multistep Method for the Solution of Third Order Ordinary Differential Equations. *General Mathematics Notes*, **25**, 62-74.

[2] Olabode, B.T. (2013) Block Multistep Method for the Direct Solution of Third Order of Ordinary Differential Equations. FUTA *Journal of Research in Sciences*, **2**, 194-200.

[3] Agboola, O. and Opanuga, A.A. (2015) Solution of Third Order Ordinary Differential Equations Using Differential Transform Method. *Global Journal of Pure and Applied Mathematics*, **11**, 2511-2516.

[4] Hashim, I. and Alshbool, M. (2019) Solving Directly Third-Order ODEs Using Operational Matrices of Bernstein Polynomials Method with Applications to Fluid Flow Equations. *Journal of King Saud University-Science*, **31**, 822-826. https://doi.org/10.1016/j.jksus.2018.05.002

[5] Lu, Y., Chen, G., Yin, Q., *et al.* (2020) Solving the Ruin Probabilities of Some Risk Models with Legendre Neural Network Algorithm. *Digital Signal Processing*, **99**, Article ID: 102634. https://doi.org/10.1016/j.dsp.2019.102634

[6] Ma, M., Zheng, L. and Yang, J. (2021) A Novel Improved Trigonometric Neural Network Algorithm for Solving Price-Dividend Functions of Continuous Time One-Dimensional Asset-Pricing Models. *Neurocomputing*, **435**, 151-161. https://doi.org/10.1016/j.neucom.2021.01.012

[7] Chen, Y., *et al*. (2021) Numerical Solving of the Generalized Black-Scholes Differential Equation Using Laguerre Neural Network. *Digital Signal Processing*, **112**, Article ID: 103003. https://doi.org/10.1016/j.dsp.2021.103003