

面向时变复数的模糊归零神经网络算法

毛华倩, 孔 颖

(浙江科技大学 信息与工程学院, 杭州 310023)

摘 要: 【目的】为了求解时变复数西尔维斯特方程(Sylvester equation), 提出两种全新的复值模糊归零神经网络(fuzzy logic system for zeroing neural network, FLSVZNN)模型。【方法】首先, 在两种复数归零神经网络(complex-valued zeroing neural network, CVZNN)模型的基础上, 引入模糊逻辑系统(fuzzy logic system, FLS)来控制信号的处理, 从而提出两种 FLSVZNN 模型; 然后, 利用李亚普诺夫定理(Lyapunov's theorem)来分析模型的稳定性和收敛速度; 最后, 通过仿真试验来进一步验证 FLSVZNN 的优越性能。【结果】在求解时变复数西尔维斯特方程时, 相比传统的神经网络模型, 使用改进的符号双幂(sign-bi-power, SBP)函数来激活的 FLSVZNN 模型具有更好的收敛性和稳定性, 可使误差函数在 0.3 s 左右收敛至 0。【结论】本研究提出的两种 FLSVZNN 模型能快速求解时变复数西尔维斯特方程, 这可为神经网络模型的建立及工程应用提供参考。

关键词: 复数归零神经网络; 模糊逻辑系统; 有限时间收敛; 激活函数

中图分类号: TP183

文献标志码: A

文章编号: 1671-8798(2024)01-0049-10

On fuzzy zeroing neural network algorithm for computing time-varying complex numbers

MAO Huaqian, KONG Ying

(School of Information and Electronic Engineering, Zhejiang University of
Science and Technology, Hangzhou 310023, Zhejiang, China)

Abstract: [Objective] In order to solve the time-varying complex Sylvester equation, two new fuzzy logic system for zeroing neural network (FLSVZNN) models were proposed. [Method] First, on the basis of two complex-valued zeroing neural network (CVZNN) models, the fuzzy logic system (FLS) was introduced to control signal processing, thus obtaining two FLSVZNN models; then, Lyapunov's theorem was used to analyze the stability and convergence rate of the models; finally, the superior performance of FLSVZNN was further verified by simulation experiments. [Result] When solving the time-varying complex Sylvester equation, compared with the traditional neural network model, the FLSVZNN models activated by the improved

收稿日期: 2023-04-23

基金项目: 浙江省自然科学基金项目(LZY22E050002)

通信作者: 孔 颖(1980—), 女, 浙江省杭州人, 教授, 博士, 主要从事神经网络与机械臂轨迹规划研究。Email: kongying-888@163.com。

sign-bi-power (SBP) function boast better convergence and stability, which can make the error function converge to 0 in 0.3 seconds. [Conclusion] Two FLSVZNN models proposed in this study can quickly solve the time-varying complex Sylvester equation, which can provide references for the establishment of neural network models and engineering applications.

Keywords: complex-valued zeroing neural network; fuzzy logic system; finite-time convergence; activation function

西尔维斯特方程(Sylvester equation)以英国数学家 James Joseph Sylvester 的名字命名,是控制理论中著名的矩阵方程^[1]。求解西尔维斯特方程一般有串行处理和并行处理两种方法,但是在很多工程领域的研究中,只有并行处理才能解决复杂的随时间变化的问题,比如图像处理、机械臂^[2]、轨迹规划、现场可编程门阵列^[3](field-programmable gate array, FPGA)、神经网络等,所以在线实时求解西尔维斯特方程对许多研究都非常重要。

随着科技的发展,与传统的方法相比,神经动力学具有更出色的性能,而且满足了在工程中能处理大量数据和实时计算的需求,因此研究者提出了多种基于梯度的递归神经网络以求解实值领域的随时间变化的西尔维斯特方程^[4]。递归神经网络(recurrent neural network, RNN)^[5-6]只保留了定义的误差函数的范数,忽视了一些随时间变化的误差信息,所以表现比较一般。基于此, Zhang 等^[7-9]提出归零神经网络(zeroing neural network, ZNN),通过对时间的一阶导数的充分利用,能有效消除随时间变化的误差信息的残差。随着对递归神经网络的研究越来越深入,在信号处理^[10]领域,涌现了越来越多需要运用复数解决的问题,比如频域在线识别或输入信号包含振幅和相位信息的情况。因此,人们开始研究实时解决复数矩阵问题的神经动力学方法^[11],从而提出了复数归零神经网络(complex-valued zeroing neural network, CVZNN)模型^[12]和各种激活函数^[13]。高畅等^[14]提出了新型有限时间归零神经网络(new finite-time zeroing neural network, NFTZNN)模型,将一种新型激活函数引入 NFTZNN 模型,从而提高收敛性能。

模糊控制将多学科,如概率论、运动学、物理学等,通过模糊集合和模糊逻辑推理转化成数字或数字函数,再用计算机实现控制^[15]。模糊逻辑系统以模糊控制为核心,应用于多个领域,如航天器、机器人^[16]、汽车等。模糊控制易于被操作人员接受,便于用计算机软件实现,鲁棒性和适应性好,而神经网络具有较高的计算效率,将模糊控制引入神经网络^[17],能代替人们在自动化和智能系统等领域中处理一些高难度的工作。

针对 CVZNN 模型求解复值矩阵问题收敛速度慢且计算精度低的问题,本研究提出加入模糊逻辑系统来求解复数时变矩阵,称之为模糊归零神经网络(fuzzy logic system for zeroing neural network, FLSVZNN)。与之前常用的收敛参数不同,输入误差函数,由模糊逻辑系统(fuzzy logic system, FLS)生成一个模糊的功率参数,并将其嵌入到模型中,从而调整收敛速度,进行理论分析与 CVZNN 模型进行比较。此外,还进行了两种实例的模拟试验,展示了两种 FLSVZNN 模型在几种常用激活函数下的优势。

1 提出问题和建立模型

1.1 问题描述

西尔维斯特方程是控制理论中的矩阵方程,形式如下:

$$\mathbf{U}(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{V}(t) - \mathbf{W}(t) = 0. \quad (1)$$

式(1)中: $\mathbf{U}(t)$ 、 $\mathbf{V}(t)$ 、 $\mathbf{W}(t)$ 为 n 阶时变复数矩阵; $\mathbf{X}(t)$ 为需求解的复数矩阵; t 为时间。

为了使上述的西尔维斯特方程成立,矩阵的行数和列数需满足一定的条件, $\mathbf{U}(t)$ 和 $\mathbf{V}(t)$ 为 n 阶和 m 阶方阵, $\mathbf{W}(t)$ 和 $\mathbf{X}(t)$ 为 n 行 m 列的矩阵。式(1)中, $m=n$,这说明 $\mathbf{X}(t)$ 有解且可以被求解。为了使 $\mathbf{X}(t)$ 有唯一解, $\mathbf{U}(t)$ 和 $-\mathbf{V}(t)$ 还需满足没有共同的特征值。

为了求解式(1),根据实域中 ZNN 模型的设计方法,建立 CVZNN 模型来求解复数时变问题。

1)构造复数的误差函数

$$\mathbf{E}(t) = \mathbf{U}(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{V}(t) - \mathbf{W}(t). \quad (2)$$

2)定义 $\mathbf{E}(t)$ 的演化过程如下:

$$\dot{\mathbf{E}}(t) = -\gamma_1 \Phi(\mathbf{E}(t)). \quad (3)$$

式(3)中: $\gamma_1 > 0$ 为可以调节的正值参数; $\Phi(\cdot)$ 为复数中的激活函数。

3)将式(2)代入式(3),可得随时间变化的西尔维斯特方程:

$$\begin{aligned} \mathbf{U}(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{V}(t) = & -\dot{\mathbf{U}}(t)\mathbf{X}(t) - \mathbf{X}(t)\dot{\mathbf{V}}(t) + \dot{\mathbf{W}}(t) - \\ & \gamma_1 \Phi(\mathbf{U}(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{V}(t) - \mathbf{W}(t)). \end{aligned} \quad (4)$$

然后在标准 ZNN 模型式(4)的基础上,将实数域扩展到复数域。

1.1.1 CVZNN-I 模型

CVZNN-I 模型的处理方法是同时激活复数输入的实部和虚部。具体描述如下:

$$\Phi_1(\mathbf{E}(t)) = \Omega(\mathbf{E}_{\text{re}}(t)) + i\Omega(\mathbf{E}_{\text{im}}(t)). \quad (5)$$

式(5)中: $\Omega(\cdot)$ 为实数的激活函数; $\mathbf{E}_{\text{re}}(t)$ 和 $\mathbf{E}_{\text{im}}(t)$ 分别为 $\mathbf{E}(t)$ 的实部和虚部。

根据式(4)和式(5)可得如下 CVZNN-I 模型:

$$\mathbf{U}(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{V}(t) = -\dot{\mathbf{U}}(t)\mathbf{X}(t) - \mathbf{X}(t)\dot{\mathbf{V}}(t) + \dot{\mathbf{W}}(t) - \gamma_1 (\Omega(\mathbf{A}(t)) + i\Omega(\mathbf{B}(t))). \quad (6)$$

式(6)中: $\mathbf{E}_{\text{re}}(t)$ 和 $\mathbf{E}_{\text{im}}(t)$ 由 $\mathbf{A}(t)$ 和 $\mathbf{B}(t)$ 来表示 $\mathbf{U}(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{V}(t) - \mathbf{W}(t)$ 的实部和虚部。

1.1.2 CVZNN-II 模型

CVZNN-II 模型的处理方法是激活复数输入的模,定义如下:

$$\Phi_2(\mathbf{E}(t)) = \Omega(\mathbf{C}(t)) \cdot \exp(i\mathbf{D}(t)). \quad (7)$$

式(7)中: $\mathbf{C}(t)$ 和 $\mathbf{D}(t)$ 分别为复数误差函数 $\mathbf{E}(t)$ 的模和辐角。

由式(4)和式(7)可得:

$$\mathbf{U}(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{V}(t) = -\dot{\mathbf{U}}(t)\mathbf{X}(t) - \mathbf{X}(t)\dot{\mathbf{V}}(t) + \dot{\mathbf{W}}(t) - \gamma_1 (\Omega(\mathbf{C}(t)) \cdot \exp(i\mathbf{D}(t))). \quad (8)$$

1.2 模糊逻辑系统

模糊逻辑是一种使用隶属度来代替传统的布尔真值的逻辑系统,与经典的二值逻辑不同,它并不使用截然不同的 0 和 1 来表达,而是使用隶属度来表达,以更好地反映现实生活中的不确定性和模糊性。这使得模糊逻辑成为一个有力的工具,可以处理那些无法被明确定义的情况,也可以在计算机中更有效地处理不确定的信息。

在传统的控制领域中,通常更精确的信息和更详细的系统建模可以提高控制系统的性能。复杂的系统由于变量太多往往很难控制,而模糊控制鲁棒性强、抗干扰能力好,在解决这类问题上表现出色。模糊逻辑系统是一种基于模糊逻辑的控制系统,它能在输入变量模糊的情况下,通过一系列模糊逻辑运算,输出模糊变量的结果。模糊系统的输入和输出通常用模糊集合来描述,模糊推理引擎中的规则和推理是模糊控制的核心。

本研究引入一个由模糊逻辑系统生成的模糊参数 v 代替之前的收敛参数,从而加速 FLSVZNN 模型的收敛过程,考虑到输入的 e 与期望的模糊参数 v 有关,制定了以下模糊逻辑方案。

第一个步骤是模糊化。隶属函数是用于表征模糊集合的数学工具,通常用来描述模糊集合的形状,将精确的输入值映射为 0 到 1 之间的模糊值。模糊化就是利用隶属函数,将输入 e 映射到模糊输入子集 E ,设定输出 v 和模糊输出子集 V 。隶属函数可以采用不同的形状,包括高斯形、三角形、梯形、S 形曲线等,本研究中模糊控制系统的输入 e 和输出 v 的隶属函数分别采用如下高斯函数和三角函数。

1)高斯函数

$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}. \quad (9)$$

式(9)中: c 为函数中心的位置; σ 为函数曲线的宽度。

2) 三角函数

$$f(x) = \begin{cases} 0, & x \leq a; \\ \frac{x-a}{b-a}, & a < x \leq b; \\ \frac{c-x}{c-b}, & b < x \leq c; \\ 0, & x > c. \end{cases} \quad (10)$$

式(10)中: $a \leq b \leq c$ 。

第二个步骤是制定模糊推理引擎。模糊推理引擎是模糊控制的核心,负责执行推理过程,将模糊输入和模糊规则转化为模糊输出,并最终将其映射为实际的操作或决策,使模糊控制能在各种领域中灵活应用。在上一步模糊化操作完成后,输入 e 转换成了模糊输出子集 E ,接下来需要明确定义一组模糊规则,这些规则描述了模糊输入子集 E 和模糊输出子集 V 之间的关系,这些关系遵循如下规则:

R_1 : 若 E 为 ZO, 则 V 为 ZO;

R_2 : 若 E 为 PS, 则 V 为 PS;

R_3 : 若 E 为 PB, 则 V 为 PB。

其中 ZO(零)、PS(正小)和 PB(正大)是模糊子集。输出参数 v 会随着输入的数值误差 e 变化, e 变大 v 也会变大, e 变小则 v 也会随之变小,以此来实现复值模糊归零神经网络的可控收敛。基于 3 个模糊规则,可得:

$$V_i = E \circ R_i. \quad (11)$$

式(11)中: \circ 为模糊变换符号; $i=1, 2, 3$, 表示 E 通过相关规则转换成 V 。具体的变换内容如下:

$$f_{E \circ R}(v) = f_{E \circ R_1}(v) \vee f_{E \circ R_2}(v) \vee f_{E \circ R_3}(v). \quad (12)$$

式(12)中: $f_{E \circ R_i}(v) = \sup(f_{E_i} \wedge f_{V_i})$, 用 \wedge 和 \vee 表示最大运算和最小运算。

第三个步骤是清晰化。为了获得最终的输出常量,需去模糊化操作,有面积中心法、面积平分法及平均最大隶属度法等。本研究采用了平均最大隶属度法:

$$v = \operatorname{argmax} f_{E \circ R}(v). \quad (13)$$

1.3 FLSVZNN 模型

引入模糊逻辑系统生成的模糊参数 v 后,设计新的随时间变化的 FLSVZNN 模型如下:

$$U(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{V}(t) = -\dot{\mathbf{U}}(t)\mathbf{X}(t) - \mathbf{X}(t)\dot{\mathbf{V}}(t) + \dot{\mathbf{W}}(t) - (\gamma_1 + \gamma_2^v)\Phi(U(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{V}(t) - \mathbf{W}(t)). \quad (14)$$

式(14)中: $r_2 > 1$ 。

然后将实值域推广到复值域,以求解复数西尔维斯特方程。通过以上方法,可以得到求解复数西尔维斯特方程的相应的两个 FLSVZNN 模型。

由式(5)和式(14),可得如下 FLSVZNN-I 模型:

$$U(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{V}(t) = -\dot{\mathbf{U}}(t)\mathbf{X}(t) - \mathbf{X}(t)\dot{\mathbf{V}}(t) + \dot{\mathbf{W}}(t) - (\gamma_1 + \gamma_2^v)(\Omega(\mathbf{A}(t)) + i\Omega(\mathbf{B}(t))). \quad (15)$$

根据式(7)和式(14),可得如下 FLSVZNN-II 模型:

$$U(t)\dot{\mathbf{X}}(t) + \dot{\mathbf{X}}(t)\mathbf{V}(t) = -\dot{\mathbf{U}}(t)\mathbf{X}(t) - \mathbf{X}(t)\dot{\mathbf{V}}(t) + \dot{\mathbf{W}}(t) - (\gamma_1 + \gamma_2^v)(\Omega(\mathbf{C}(t)) \cdot \exp(i\mathbf{D}(t))). \quad (16)$$

1.4 激活函数

越来越多的激活函数被用来改善神经网络的收敛性能。

1) 线性函数

$$\Omega(x) = x. \quad (17)$$

2) 功率函数

$$\Omega(x) = x^k, k \text{ 为奇数}, k \geq 3. \quad (18)$$

3) 双相符号函数

$$\Omega(x) = \frac{1 + e^{-m}}{1 - e^{-m}} \frac{1 - e^{-mx}}{1 + e^{-mx}}, m > 2. \quad (19)$$

4) 符号双幂(sign-bi-power, SBP) 函数

$$\Omega(x) = \frac{1}{2} |x|^m \text{sign}(x) + \frac{1}{2} |x|^{\frac{1}{m}} \text{sign}(x). \quad (20)$$

式(20)中: $m \in (0, 1)$, $x \in \mathbb{R}$ 。

5) 改进的 SBP 函数

$$\Omega(x) = \frac{1}{2} |x|^m \text{sign}(x) + \frac{1}{2} x + \frac{1}{2} |x|^{\frac{1}{m}} \text{sign}(x). \quad (21)$$

与 SBP 函数相比,改进的 SBP 函数有一个额外的线性项。因此,在 FLSVZNN-I 模型和 FLSVZNN-II 模型中,使用改进的 SBP 函数来激活比使用 SBP 函数来激活效果更好。

2 收敛性能分析

定理 1 给定光滑的时变复矩阵 $\mathbf{U}(t)$ 、 $\mathbf{V}(t)$ 、 $\mathbf{W}(t)$,当使用激活函数 $\Omega(x) = x$ 时,FLSVZNN-I 模型的矩阵 $\mathbf{X}(t)$ 能从随机原始状态 $\mathbf{X}(0)$ 收敛到时变复数西尔维斯特方程的理论解 $\mathbf{X}^*(t)$ 。

证明:根据式(3)、式(5)和式(15)可知 $\mathbf{A}(t)$ 和 $\mathbf{B}(t)$ 等价于以下两个方程:

$$\begin{cases} \dot{\mathbf{A}}(t) = -(\gamma_1 + \gamma_2^v) \Omega(\mathbf{A}(t)); \\ \dot{\mathbf{B}}(t) = -(\gamma_1 + \gamma_2^v) \Omega(\mathbf{B}(t)). \end{cases} \quad (22)$$

将 $\dot{\mathbf{A}}(t)$ 、 $\dot{\mathbf{B}}(t)$ 用元素来表示,即

$$\begin{cases} \dot{a}_{ij}(t) = -(\gamma_1 + \gamma_2^v) \Omega(a_{ij}(t)); \\ \dot{b}_{ij}(t) = -(\gamma_1 + \gamma_2^v) \Omega(b_{ij}(t)). \end{cases} \quad (23)$$

式(23)中: $a_{ij}(t)$ 为 $\mathbf{A}(t)$ 中第 i 行 j 列的值。定义李亚普诺夫函数 $L = \frac{1}{2} a_{ij}^2(t)$, 可得 $\dot{L} = a_{ij}(t) \dot{a}_{ij}(t) = -(\gamma_1 + \gamma_2^v) a_{ij}(t) \Omega(a_{ij}(t))$ 。由于激活函数 $\Omega(x) = x$ 是单调递增的奇函数,故满足 $\Omega(a_{ij}(t)) = -\Omega(-a_{ij}(t))$ 。因为 $\gamma_1 > 0$, $\gamma_2 > 1$, $v > 0$, 所以 $\gamma_1 + \gamma_2^v > 0$ 。当 $a_{ij}(t) \geq 0$ 时,因为 $\Omega(x) = x$, 所以 $\Omega(a_{ij}(t)) \geq 0$, 李亚普诺夫函数的导数 $\dot{L} \leq 0$; 当 $a_{ij}(t) < 0$ 时, $\Omega(a_{ij}(t)) < 0$, $\dot{L} < 0$ 。 $b_{ij}(t)$ 的结果同理可得。

因此,对于误差函数 $\mathbf{A}(t)$ 和 $\mathbf{B}(t)$ 任意 i 行 j 列的值, \dot{L} 恒小于等于 0。那么根据李亚普诺夫定理,这两个动态系统将收敛到它们的平衡点,即 $\mathbf{A}(t)$ 和 $\mathbf{B}(t)$ 都随时间收敛至 0, 可得误差函数 $\mathbf{E}(t)$ 能最终全局收敛。根据 $\mathbf{E}(t)$ 的定义,FLSVZNN-I 模型能全局稳定收敛,被求解的矩阵 $\mathbf{X}(t)$ 能从随机原始状态 $\mathbf{X}(0)$ 收敛至理论解 $\mathbf{X}^*(t)$ 。

定理 2 给定光滑的时变复矩阵 $\mathbf{U}(t)$ 、 $\mathbf{V}(t)$ 、 $\mathbf{W}(t)$,当使用激活函数 $\Omega(x) = x$ 时,FLSVZNN-II 模型的矩阵 $\mathbf{X}(t)$ 能从随机原始状态 $\mathbf{X}(0)$ 收敛到时变复数西尔维斯特方程的理论解 $\mathbf{X}^*(t)$ 。

从 FLSVZNN-II 模型中可以得到如下非线性动态子函数:

$$\dot{e}_{ij}(t) = -(\gamma_1 + \gamma_2^v) (\Omega(c_{ij}(t)) \cdot \exp(\text{id}_{ij}(t))). \quad (24)$$

式(24)中: $\dot{e}_{ij}(t)$ 、 $c_{ij}(t)$ 、 $d_{ij}(t)$ 分别为矩阵函数 $\dot{\mathbf{E}}(t)$ 、 $\mathbf{C}(t)$ 和 $\mathbf{D}(t)$ 第 i 行 j 列的值; $\Omega(\cdot)$ 中 $c_{ij}(t)$ 和 $d_{ij}(t)$ 分别为复数变量 $e_{ij}(t)$ 的模和辐角。

为了证明上述系统的稳定性,选择李亚普诺夫函数 $L = \frac{1}{2} |e_{ij}(t)|^2 = \frac{1}{2} c_{ij}^2(t)$ 。因为 $L = \frac{1}{2} |e_{ij}(t)|^2 = \frac{1}{2} e_{ij}(t) \overline{e_{ij}(t)}$, 所以对 L 进行求导,结合 $\overline{e_{ij}(t)} = c_{ij}(t) \exp(-\text{id}_{ij}(t))$, $\overline{\dot{e}_{ij}(t)} = -(\gamma_1 + \gamma_2^v) \Omega(c_{ij}(t)) \cdot \exp(-\text{id}_{ij}(t))$, 可得:

$$\dot{L} = \frac{1}{2} (\dot{e}_{ij}(t) \overline{e_{ij}(t)} + e_{ij}(t) \overline{\dot{e}_{ij}(t)}) = -(\gamma_1 + \gamma_2^v) c_{ij}(t) \Omega(c_{ij}(t)) = -(\gamma_1 + \gamma_2^v) |e_{ij}(t)| \Omega(|e_{ij}(t)|).$$

考虑到 $\Omega(x) = x$ 是一个单调递增的奇函数, $\gamma_1 > 0$, $\gamma_2 > 1$, $v > 0$, 可得 $\gamma_1 + \gamma_2^v > 0$ 。那么当 $e_{ij}(t) \geq 0$

时, 因为 $\Omega(x) = x$, 所以 $\Omega(e_{ij}(t)) \geq 0$, 李亚普诺夫函数的导数 $\dot{L} \leq 0$; 当 $e_{ij}(t) < 0$ 时, $\Omega(e_{ij}(t)) < 0, \dot{L} < 0$ 。可知, \dot{L} 恒小于等于 0。根据李亚普诺夫理论, 对于误差函数 $\mathbf{E}(t)$ 任意 i 行 j 列的值, 都能全局收敛, 即误差函数 $\mathbf{E}(t)$ 最终能全局收敛。根据 $\mathbf{E}(t)$ 的定义, FLSVZNN-II 模型能全局稳定收敛, 被求解的矩阵 $\mathbf{X}(t)$ 能从随机原始状态 $\mathbf{X}(0)$ 收敛至理论解 $\mathbf{X}^*(t)$ 。

定理 3 给定光滑的时变复矩阵 $\mathbf{U}(t)$ 、 $\mathbf{V}(t)$ 、 $\mathbf{W}(t)$, 当使用激活函数 $\Omega(x) = x$ 时, FLSVZNN-I 模型的收敛速度为 $\gamma_1 + \gamma_2^v$, 比 CVZNN-I 模型收敛速度更快。

$q(t)$ 记录 $a_{ij}(t)$ 和 $b_{ij}(t)$ 的最大绝对值, 当 $t = 0$ 时, 可以取得较大的初始值, 记为 $q(0) = \max\{|a_{ij}(0)|, |b_{ij}(0)|\}$ 。此外, 可以得到 $-|q(t)| \leq |a_{ij}(t)| \leq |q(t)|$ 和 $-|q(t)| \leq |b_{ij}(t)| \leq |q(t)|$ 。换言之, 当 $q(t) = 0$ 时, $\mathbf{A}(t)$ 和 $\mathbf{B}(t)$ 中的所有元素都可以达到 0。则 $q(t)$ 随时间变化表现为 $\dot{q}(t) = -(\gamma_1 + \gamma_2^v)\Omega(q(t))$ 。

定义了一个李亚普诺夫函数 $L(t) = |q(t)|^2$, 并求解了 $L(t)$ 随时间变化的时间导数, 因为使用 $\Omega(x) = x$, 可得 $\dot{L}(t) = -2(\gamma_1 + \gamma_2^v)q(t)^2 = -2(\gamma_1 + \gamma_2^v)L(t)$ 。由上式可得 $\frac{1}{L(t)}dL(t) = -2(\gamma_1 + \gamma_2^v)dt$, 求解上述等式, 可以得到 $L(t) = L(0) \cdot e^{-2(\gamma_1 + \gamma_2^v)t}$, 其中 $L(0) = |q(0)|^2$, 因此 $q(t) = q(0) \cdot e^{-(\gamma_1 + \gamma_2^v)t}$ 。由于

$$\|q(t)\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m q(t)^2}, \text{ 可知 } \|q(t)\|_F = \|q(0)\|_F \cdot e^{-(\gamma_1 + \gamma_2^v)t}.$$

一般情况下, 可以得出具有任意初值的 FLSVZNN-I 模型的收敛速度为 $\gamma_1 + \gamma_2^v$ 。同理可得, 当没有使用由模糊逻辑系统产生的模糊参数 v 时, 使用参数为 γ_1 的 CVZNN-I 模型的收敛速度为 γ_1 。因为 $\gamma_1 > 0, \gamma_2 > 1, v > 0$, 所以 $\gamma_1 + \gamma_2^v > \gamma_1$, 故 FLSVZNN-I 模型比 CVZNN-I 模型收敛速度更快。

定理 4 给定光滑的时变复矩阵 $\mathbf{U}(t)$ 、 $\mathbf{V}(t)$ 、 $\mathbf{W}(t)$, 当使用激活函数 $\Omega(x) = x$ 时, FLSVZNN-II 模型的收敛速度为 $\gamma_1 + \gamma_2^v$, 比 CVZNN-II 模型收敛速度更快。

计算 FLSVZNN-II 模型收敛速度的方法与 FLSVZNN-I 模型的方法相似。首先, 定义李亚普诺夫函数 $L(t) = |u(t)|^2$, 其中 $u(t)$ 满足 $|u(t)| = \max|e_{ij}(t)| = \max|c_{ij}(t)|$ 。显然, 当 $u(t) = 0$ 时, $|u(0)| = \max|e_{ij}(0)| = \max|c_{ij}(0)|$, $\mathbf{E}(t)$ 中的所有元素都可以收敛至 0。通过计算 $u(t)$ 的收敛速度, 可以预测 FLSVZNN-II 模型的收敛速度。 $u(t)$ 的动态演化如下: $\dot{u}(t) = -(\gamma_1 + \gamma_2^v)\Omega(u(t))$ 。

根据计算, 可以进一步得到 $\dot{L}(t) = -2(\gamma_1 + \gamma_2^v)u(t)\Omega(u(t)) = -2(\gamma_1 + \gamma_2^v)u^2(t)$ 。然后, 类似于 FLSVZNN-I 模型的证明, 可以得出具有任意初值的 FLSVZNN-II 模型的收敛速度为 $\gamma_1 + \gamma_2^v$ 。同理可得, 当没有使用由模糊逻辑系统产生的模糊参数 v 时, 使用参数为 γ_1 的 CVZNN-II 模型的收敛速度为 γ_1 。因为 $\gamma_1 > 0, \gamma_2 > 1, v > 0$, 所以 $\gamma_1 + \gamma_2^v > \gamma_1$, 故 FLSVZNN-II 模型比 CVZNN-II 模型收敛速度更快。

3 模拟验证

以下给出两个具体的例子, 利用激活函数 $\Omega(x) = x$, 验证 FLSVZNN-I 模型和 FLSVZNN-II 模型的有效性。

例 1 考虑了具有以下系数 $\mathbf{U}(t)$ 、 $\mathbf{V}(t)$ 、 $\mathbf{W}(t)$ 和 $\gamma_1 = 1, \gamma_2 = 5$ 的复值时, 变西尔维斯特方程

$$\mathbf{U}(t) = \begin{bmatrix} e^{5it} & 0 \\ e^{5it} & \sin(5t) + \cos(5t)i \end{bmatrix}, \mathbf{V}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{W}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (25)$$

求解可知 $\mathbf{U}(t)$ 的特征值 $\lambda_1 = e^{5it}, \lambda_2 = \sin(5t) + \cos(5t)i$; $\mathbf{V}(t)$ 的特征值 $\lambda_1 = 0, \lambda_2 = 0$ 。 $\mathbf{U}(t)$ 和 $-\mathbf{V}(t)$ 没有共同的特征值, 满足 $\mathbf{X}(t)$ 是唯一解。

使用 FLSVZNN-I 模型和 FLSVZNN-II 模型计算例 1 的状态轨迹, 分别如图 1 和图 2 所示, 可知这两种模型在激活函数 $\Omega(x) = x$ 的激活作用下, 都可以从一个特定的初始状态 $\mathbf{X}(0) = [2, 2; -2, 2]$ 开始, 使复数矩阵 $\mathbf{X}(t)$ 从用实心蓝线表示的状态解收敛到用红色虚线表示的理论解, 有较好的收敛性能。

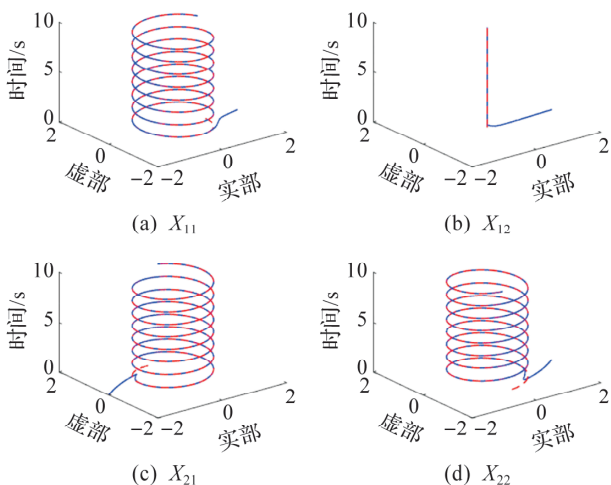


图 1 使用 FLSVZNN-I 模型计算例 1 的状态轨迹

Fig. 1 State trajectories of Example 1 computed by FLSVZNN-I model

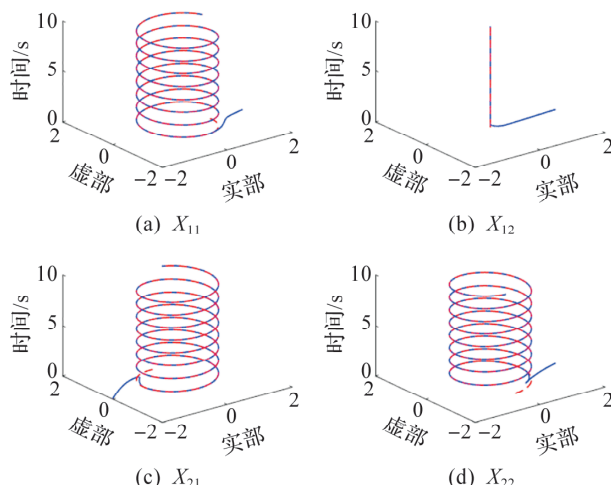


图 2 使用 FLSVZNN-II 模型计算例 1 的状态轨迹

Fig. 2 State trajectories of Example 1 computed by FLSVZNN-II model

FLSVZNN-I 模型、FLSVZNN-II 模型使用不同激活函数计算例 1 时产生的不同误差收敛时间分别如图 3 和图 4 所示。可知,当使用多种激活函数来激活 FLSVZNN 模型时,尽管 FLSVZNN-I 模型和 FLSVZNN-II 模型存在差异,但都能在不同的激活函数下使误差函数收敛,这验证了两种处理方法求解复数西尔维斯特方程的有效性。此外,由图 3 和图 4 可知,当 FLSVZNN-I 模型使用线性函数激活时,计算误差约 1.0 s 左右收敛至 0;使用功率函数时,无法使其收敛到 0;使用双相符号函数激活时,计算误差约 1.0 s 左右收敛至 0;使用 SBP 函数为激活函数时,计算误差至少 0.4 s 左右收敛至 0;使用改进的 SBP 激活函数时,计算误差约 0.3 s 左右收敛至 0。FLSVZNN-II 模型的结果类似。可以得出结论,在收敛速度方面,使用改进的 SBP 激活函数收敛速度最快。

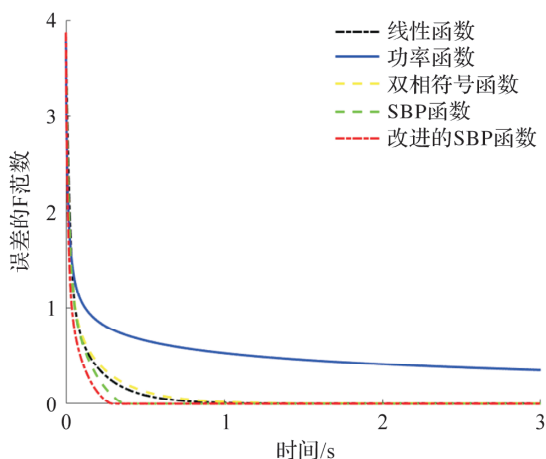


图 3 FLSVZNN-I 模型使用不同激活函数计算例 1 时产生的不同误差收敛时间

Fig. 3 Convergence time of different errors for Example 1 calculated by FLSVZNN-I model using different activation functions

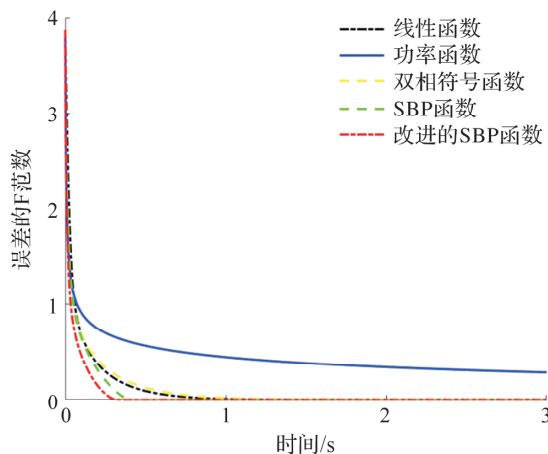


图 4 FLSVZNN-II 模型使用不同激活函数计算例 1 时产生的不同误差收敛时间

Fig. 4 Convergence time of different errors for Example 1 calculated by FLSVZNN-II model using different activation functions

图 5 和图 6 分别显示了使用 FLSVZNN 模型和 CVZNN 模型计算例 1 的收敛时间对比,结果进一步验证了 FLSVZNN-I 模型和 FLSVZNN-II 模型在使用改进的 SBP 激活函数作用下处理复数矩阵的有效性。由图 5 可知,FLSVZNN-I 模型求解的收敛时间约 0.3 s;而 CVZNN-I 模型求解的收敛时间至少 2.2 s。可见在收敛速度方面,FLSVZNN-I 模型明显优于 CVZNN-I 模型。FLSVZNN-II 模型的结果类似。这验证了在 CVZNN 神经网络加入模糊逻辑系统产生的模糊参数 v 后,生成的两个 FLSVZNN 模型性能更优。

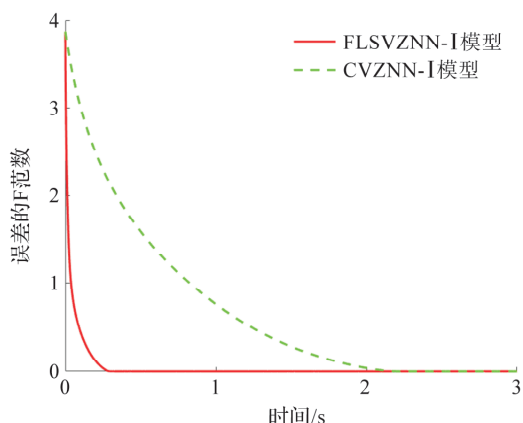


图 5 使用 FLSVZNN-I 模型和 CVZNN-I 模型计算例 1 的收敛时间对比

Fig. 5 Comparison of convergence time of Example 1 calculated by FLSVZNN-I model and CVZNN-I model

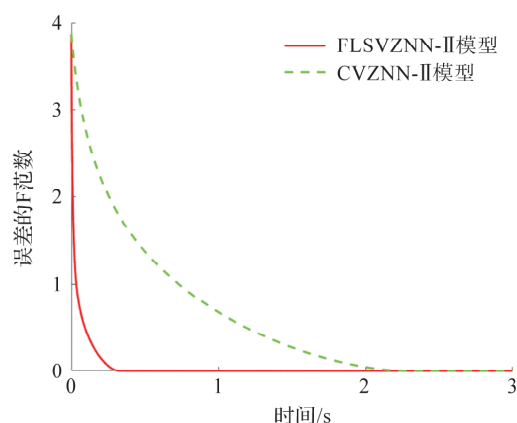


图 6 使用 FLSVZNN-II 模型和 CVZNN-II 模型计算例 1 的收敛时间对比

Fig. 6 Comparison of convergence time of Example 1 calculated by FLSVZNN-II model and CVZNN-II model

例 2 考虑了具有系数 $U(t)$ 、 $V(t)$ 、 $W(t)$ 的复数时变西尔维斯特方程:

$$U(t) = \begin{bmatrix} e^{it} & -ie^{-it} \\ -e^{it} & e^{-it} \end{bmatrix}, V(t) = \begin{bmatrix} -2ie^{2it} & e^{-it} \\ e^{it} & -2ie^{-2it} \end{bmatrix}, W(t) = \begin{bmatrix} -ie^{it} + \frac{1}{2}i & \frac{e^{-2it}}{2} + e^{-3it} \\ \frac{e^{2it}}{2} + e^{3it} & -ie^{-it} - i \end{bmatrix}. \quad (26)$$

求解可知: 例 2 的 $U(t)$ 的特征值 $\lambda_1 = \frac{e^{-it}(e^{2it} - (-2e^{2it} + e^{2it} + (1+4i))^{0.5} + 1)}{2}$, $\lambda_2 = \frac{e^{-it}(e^{2it} + (-2e^{2it} + e^{2it} + (1+4i))^{0.5} + 1)}{2}$; $V(t)$ 的特征值 $\lambda_1 = -2ie^{2it} - 1$, $\lambda_2 = 1 - 2ie^{2it}$ 。 $U(t)$ 和 $-V(t)$ 同样没有共同的特征值, 满足 $X(t)$ 是唯一解。

使用 FLSVZNN-I 模型和 FLSVZNN-II 模型求解例 2 的状态轨迹, 分别如图 7 和图 8 所示。

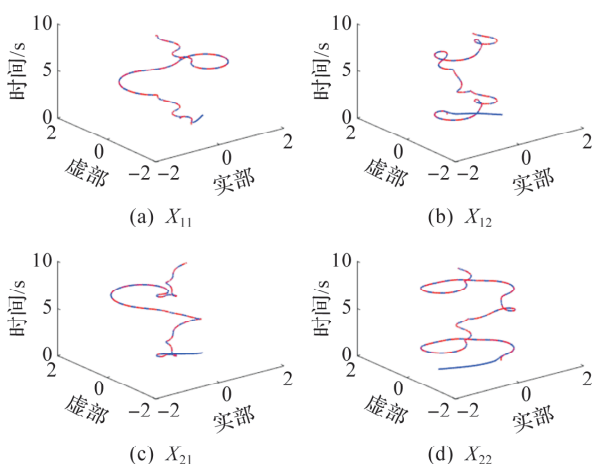


图 7 使用 FLSVZNN-I 模型计算例 2 的状态轨迹
Fig. 7 State trajectories of Example 2 computed by FLSVZNN-I model

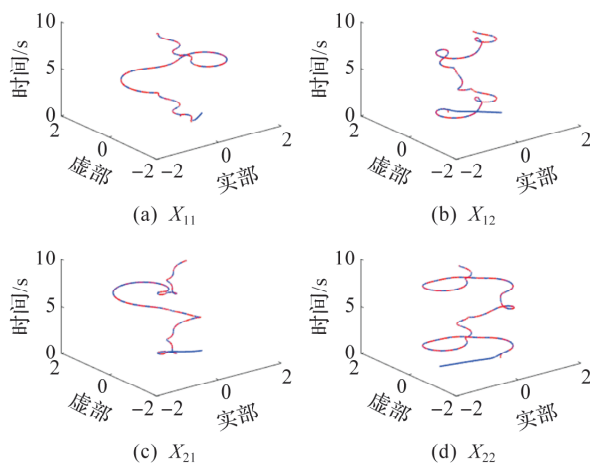


图 8 使用 FLSVZNN-II 模型计算例 2 的状态轨迹
Fig. 8 State trajectories of Example 2 computed by FLSVZNN-II model

由图 7 和图 8 可知, 这两种模型都可以在激活函数 $\Omega(x) = x$ 下, 从一个特定的初始状态 $X(0) = [1, 1; 1, -1]$ 开始, 使复数矩阵 $X(t)$ 从用实心蓝线表示的状态解收敛到用红色虚线表示的理论解, 有较好的收敛性能。

图 9 和图 10 分别为 FLSVZNN-I 模型和 FLSVZNN-II 模型使用不同激活函数计算例 2 所产生的不同误差收敛时间。由图 9 和图 10 可知, 当使用多种激活函数来激活 FLSVZNN 模型时, 尽管

FLSVZNN-I 模型和 FLSVZNN-II 模型存在差异,但都能在不同的激活函数下使误差函数收敛,这进一步验证了两种处理方法求解复数西尔维斯特方程的有效性。此外,由图 9 和图 10 还知,当 FLSVZNN-I 模型使用线性函数激活时,计算误差约 0.8 s 左右收敛至 0;使用功率函数时,无法使其收敛至 0;使用双相符号函数激活时,计算误差约 0.8 s 左右收敛至 0;使用 SBP 函数为激活函数时,计算误差约 0.3 s 左右收敛至 0;使用改进的 SBP 激活函数时,计算误差约 0.3 s 左右收敛至 0。FLSVZNN-II 模型的结果类似。可见,在收敛速度方面,使用改进的 SBP 激活函数最快。

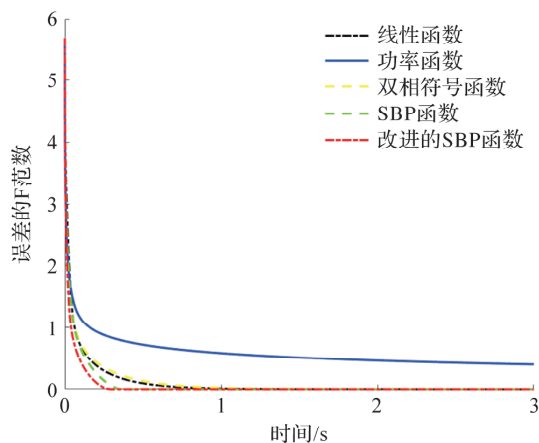


图 9 FLSVZNN-I 模型使用不同激活函数计算例 2 所产生的不同误差收敛时间

Fig. 9 Convergence time of different errors for Example 2 calculated by FLSVZNN-I model using different activation functions

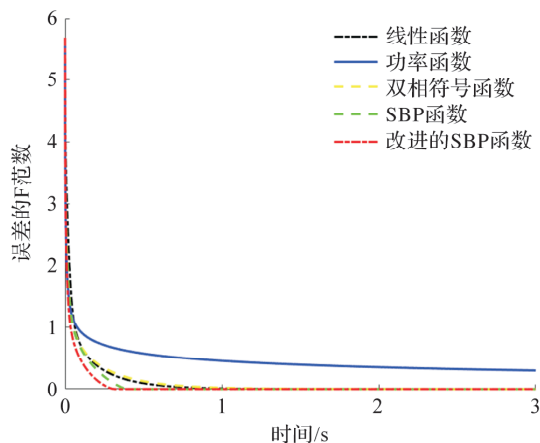


图 10 FLSVZNN-II 模型使用不同激活函数计算例 2 所产生的不同误差收敛时间

Fig. 10 Convergence time of different errors for Example 2 calculated by FLSVZNN-II model using different activation functions

图 11 和图 12 分别显示了使用 FLSVZNN 模型和 CVZNN 模型计算例 2 的收敛时间对比。

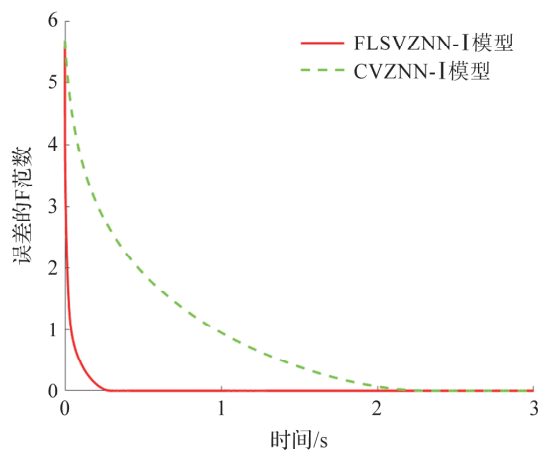


图 11 使用 FLSVZNN-I 模型和 CVZNN-I 模型计算例 2 的收敛时间对比

Fig. 11 Comparison of convergence time of Example 2 calculated by FLSVZNN-I model and CVZNN-I model

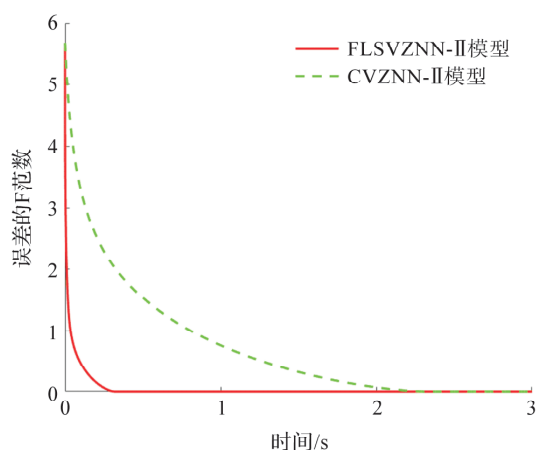


图 12 使用 FLSVZNN-II 模型和 CVZNN-II 模型计算例 2 的收敛时间对比

Fig. 12 Comparison of convergence time of Example 2 calculated by FLSVZNN-II model and CVZNN-II model

图 11 和图 12 的结果进一步验证了 FLSVZNN-I 模型和 FLSVZNN-II 模型在改进的 SBP 激活函数作用下处理复数矩阵的有效性。由图 11 可知,FLSVZNN-I 模型求解的收敛时间约 0.3 s;而 CVZNN-I 模型求解的收敛时间至少 2.3 s。可见在收敛速度方面,FLSVZNN-I 模型明显优于 CVZNN-I 模型。FLSVZNN-II 模型的结果类似。这验证了在 CVZNN 神经网络加入模糊逻辑系统产生的模糊参数 v 后,两个 FLSVZNN 模型相对于两个 CVZNN 模型性能更优。

FLSVZNN 模型在不同激活函数作用下的收敛时间见表 1。

表 1 FLSVZNN 模型在不同激活函数作用下的收敛时间

Table 1 Convergence time of FLSVZNN models under different activation functions

s

例子	神经网络	线性函数	功率函数	双相符号函数	SBP 函数	改进的 SBP 函数
例 1	FLSVZNN-I 模型	1.0	不收敛	1.0	0.4	0.3
	FLSVZNN-II 模型	1.0	不收敛	1.0	0.4	0.3
例 2	FLSVZNN-I 模型	0.8	不收敛	0.8	0.4	0.3
	FLSVZNN-II 模型	0.8	不收敛	0.8	0.4	0.3

4 结 语

本研究针对求解时变复数西尔维斯特方程的问题,提出了两种新型的复值模糊神经网络。根据李亚普诺夫定理,对这两种 FLSVZNN 模型的收敛性能进行了理论分析,并计算了 FLSVZNN 模型和 CVZNN 模型的收敛速度,验证了 FLSVZNN 模型求解的有效性和可行性。最后由数值仿真可知,与传统的激活函数相比,使用改进的 SBP 函数来激活的两种 FLSVZNN 模型,收敛速度更快。

参考文献:

- [1] XIAO L, LIAO B, LI S, et al. Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations[J]. Neural Networks, 2018, 98: 102.
- [2] XIAO L, ZHANG Z, ZHANG Z, et al. Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation[J]. Neural Networks, 2018, 105: 185.
- [3] DAHIPHALE V, BANSOD G, ZAMBARE A, et al. Design and implementation of various datapath architectures for the ANU lightweight cipher on an FPGA[J]. Frontiers of Information Technology and Electronic Engineering, 2020, 21(4): 615.
- [4] SHEN Y, MIAO P, HUANG Y, et al. Finite-time stability and its application for solving time-varying Sylvester equation by recurrent neural network[J]. Neural Processing Letters, 2015, 42(3): 767.
- [5] WANG J. Recurrent neural networks for computing pseudoinverses of rank-deficient matrices[J]. SIAM Journal on Scientific Computing, 1997, 18(5): 1479.
- [6] WANG X, CHE M, WEI Y. Recurrent neural network for computation of generalized eigenvalue problem with real diagonalizable matrix pair and its applications[J]. Neurocomputing, 2016, 216: 230.
- [7] ZHANG Z, KONG L, ZHENG L, et al. Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying QM and QP problems and applications[J]. IEEE Transactions on Systems, Man, and Cybernetics; Systems, 2020, 50(12): 5106.
- [8] LI W, MA X, LUO J, et al. A strictly predefined-time convergent neural solution to equality- and inequality-constrained time-variant quadratic programming [J]. IEEE Transactions on Systems, Man, and Cybernetics; Systems, 2021, 51(7): 4028.
- [9] YU F, LIU L, XIAO L, et al. A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function[J]. Neurocomputing, 2019, 350: 108.
- [10] PANDE A, GOEL V. Complex-valued neural network in image recognition: a study on the effectiveness of radial basis function[J]. World Academy of Science, Engineering and Technology, 2007, 26: 220.
- [11] ZHANG Y, ZHAN L, LI K. Complex-valued Zhang neural network for online complex-valued time-varying matrix inversion[J]. Applied Mathematics & Computation, 2011, 217(24): 10066.
- [12] 孔颖, 孙明轩. Sylvester 时变矩阵方程求解的终态神经网络算法[J]. 计算机科学, 2018, 45(10): 6.
- [13] LI S, CHEN S, LIU B. Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function[J]. Neural Processing Letters, 2013, 37(2): 189.
- [14] 高畅, 孔颖, 胡汤珑. 基于有限时间神经网络求解的时变复数矩阵方程[J]. 浙江科技学院学报, 2022, 34(5): 409.
- [15] 石辛民, 郝整清. 模糊控制及其 MATLAB 仿真[M]. 北京: 清华大学出版社, 2008.
- [16] 孟廷豪. 基于模糊控制的机器人避障研究[D]. 太原: 中北大学, 2013.
- [17] ZHANG Z, YAN Z. An adaptive fuzzy recurrent neural network for solving the nonrepetitive motion problem of redundant robot manipulators[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(4): 684.