

# 梯度神经网络方法求解含有绝对值形式张量特征值

蔡泽福

云南师范大学数学学院, 云南 昆明

收稿日期: 2024年8月24日; 录用日期: 2024年9月18日; 发布日期: 2024年9月27日

## 摘要

目前, 关于特征值的研究主要集中在特征值互补、特征值估计和运用算法计算特征值等方向。受张量绝对值方程  $\mathcal{A}x^{m-1} - |x| = b$  启示, 本文考虑一类新形式的特征值问题, 并提出梯度神经网络方法求解新形式张量特征值和特征向量。数值实验表明了梯度神经网络方法求解该问题的可行性和有效性。

## 关键词

张量, 特征值, 梯度神经网络

# Gradient Neural Network Method for Solving Tensor Eigenvalues with Absolute Value Form

Zefu Cai

School of Mathematics, Yunnan Normal University, Kunming Yunnan

Received: Aug. 24<sup>th</sup>, 2024; accepted: Sep. 18<sup>th</sup>, 2024; published: Sep. 27<sup>th</sup>, 2024

## Abstract

At present, research on eigenvalues mainly focuses on complementary eigenvalues, eigenvalue estimation, and the application of algorithms to calculate eigenvalues. Inspired by the tensor absolute value equation  $\mathcal{A}x^{m-1} - |x| = b$ , this paper considers a new form of eigenvalue problem and proposes a gradient neural network method to solve the eigenvalues and eigenvectors of the new form tensor. Numerical experiments have shown the feasibility and effectiveness of using gradient neural

network methods to solve this problem.

## Keywords

Tensor, Eigenvalue, Gradient Neural Network

Copyright © 2024 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. 介绍

2005年, Qi在[1]中提出了张量特征值的概念, 即, 令  $\mathcal{A} = (a_{i_1 i_2 \dots i_m})$  是一个  $m$  阶  $n$  维张量, 若存在一个数  $\lambda \in \mathbb{C}$  和非零向量  $x \in \mathbb{C}^n$  满足

$$\mathcal{A}x^{m-1} = \lambda x^{[m-1]},$$

则  $\lambda$  叫做张量  $\mathcal{A}$  的特征值,  $x$  叫做  $\lambda$  所对应的特征向量, 其中  $\mathcal{A}x^{m-1}$  和  $x^{[m-1]}$  都表示一个  $n$  维向量, 且

$$\begin{aligned} (\mathcal{A}x^{m-1})_i &= \sum_{i_2 \dots i_m} a_{i i_2 \dots i_m} x_{i_2} \cdots x_{i_m}, \forall i = \{1, 2, \dots, n\}, \\ (x^{[m-1]})_i &= x_i^{m-1}. \end{aligned}$$

若  $\lambda$  和  $x$  都属于实数域, 则  $(\lambda, x)$  叫做张量  $\mathcal{A}$  的 H-特征对。此外, Qi也在[1]中定义了 Z-特征值, 即, 若存在数  $\lambda \in \mathbb{R}$  和向量  $x \in \mathbb{R}^n$  满足

$$\begin{cases} \mathcal{A}x^{m-1} = \lambda x, \\ x^T x = 1, \end{cases} \quad (1.1)$$

则称特征对  $(\lambda, x)$  是 Z-特征对。值得注意的是,  $\lambda$  和  $x$  属于复数域时, 称特征对  $(\lambda, x)$  为 E-特征对。

张量 Z-特征值在许多实际问题中都被运用, 如自动控制[2], 人脑科学[3], 统计数据分析中的最佳秩一逼近[4]等领域。为此, 很多学者对张量 Z-特征值的求解展开了研究。

目前, 已经有许多计算对称或非负张量 Z-特征对的算法, 但是结合[5]中的张量绝对值方程

$$\mathcal{A}x^{m-1} - |x| = b, \quad (1.2)$$

以及张量 Z-特征值的一般形式(1.1), 考虑形式为

$$\begin{cases} \mathcal{A}x^{m-1} = \lambda |x|, \\ x^T x = 1, \end{cases} \quad (1.3)$$

的特征值问题。其中  $\mathcal{A}$  是非奇异张量,  $|x| = (|x_1|, |x_2|, \dots, |x_n|)^T$ , 受[6]启示, (1.3)可改写成

$$\begin{cases} \mathcal{A}x^{m-1} = \lambda D x, \\ x^T x = 1, \end{cases}$$

其中  $D = (d_{ij}) \in \mathbb{R}^{n \times n}$ , 是一个对角矩阵, 其主对角线元素  $d_{ii}$  满足

$$d_{ii} = \begin{cases} 1, & x_i > 0, \\ 0, & x_i = 0, \\ -1, & x_i < 0. \end{cases}$$

该问题涉及到绝对值, 其在原点处是不可微的。据我们所知该问题还未研究, 这便是本文的研究意义。

求解(1.3)中  $\lambda$  和  $x$  的过程实际上是求解约束化非线性方程组的问题。非线性方程组的求解方法有许多, 比如说梯度投影法、罚函数法等。在本文, 我们用梯度神经网络方法求解含有绝对值形式的张量特征值和特征向量。

## 2. 构造梯度神经网络模型

梯度神经网络现在已经被认为是数值计算中的一个强大的算法[4], 在矩阵求逆和 Drazin 逆[5], 线性和非线性方程组[6]的求解等领域有着十分重要的作用。鉴于其高速处理特性和在实际应用中硬件实现的方便性, 更多梯度神经网络求解线性方程组与非线性方程组问题见[7]-[10]。

为了监测和控制(1.3)的求解过程, 根据梯度神经网络方法设计思路[11], 首先定义一个误差函数

$$E(x(t), \lambda(t)) := \mathcal{A}x(t)^{m-0} - \lambda(t)Dx(t) \in \mathbb{R}^n. \quad (2.1)$$

为了迫使  $E(x(t), \lambda(t))$  收敛于零。类似于[12]的工作, 接着定义一个监测误差函数

$$e(x(t), \lambda(t)) := \frac{1}{2} \|\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)\|_2^2 \in \mathbb{R}.$$

显然, 当监测误差函数  $e(x(t), \lambda(t))$  收敛于零时,  $E(x(t), \lambda(t))$  也收敛于零。

为了实现  $E(x(t), \lambda(t))$  收敛到零的目的, 根据梯度神经网络设计方法, 监测误差函数的负梯度方向作为下降方向, 即

$$\begin{cases} \frac{dx}{dt} = -\gamma_1 \frac{\partial e(x(t), \lambda(t))}{\partial x}, \\ \frac{d\lambda}{dt} = -\gamma_2 \frac{\partial e(x(t), \lambda(t))}{\partial \lambda}. \end{cases} \quad (2.2)$$

把(2.2)展开, 有

$$\begin{cases} \frac{dx}{dt} = -\gamma_1 ((m-1)\mathcal{A}x^{m-2} - \lambda DI)^T (\mathcal{A}x^{m-1} - \lambda Dx), \\ \frac{d\lambda}{dt} = \gamma_2 (Dx)^T (\mathcal{A}x^{m-1} - \lambda Dx), \end{cases} \quad (2.3)$$

其中  $\gamma_1 > 0$  和  $\gamma_2 > 0$ 。

采用欧拉差分对(2.3)进行离散, 得

$$\begin{cases} x_{k+1} = x_k - \alpha_1 ((m-1)\mathcal{A}x_k^{m-2} - \lambda_k D_k I)^T (\mathcal{A}x_k^{m-1} - \lambda_k D_k x_k) = x_k - f(x_k, \lambda_k), \\ \lambda_{k+1} = \lambda_k + \alpha_2 (D_k x_k)^T (\mathcal{A}x_k^{m-1} - \lambda_k D_k x_k) = \lambda_k + g(x_{k+1}, \lambda_k), \end{cases} \quad (2.4)$$

其中  $\tau_i$  是步长,  $\alpha_i = \tau_i \gamma_i (i=1,2)$ ,  $\tau_i$  应足够小。

故获得了一个计算(1.3)中  $\lambda$  和  $x$  的算法步骤, 如下

步骤一: 给定一个张量  $\mathcal{A} \in S^{[m,n]}$ , 误差参数  $\varepsilon > 0$ , 最大代步数  $k_{\max}$  和初始向量  $x_0 \in \Sigma = \{x | x^T x = 1\}$ ;

步骤二: 令  $\alpha_1, \alpha_2 > 0$ ,  $\lambda_0 \in \mathbb{R}$ ,  $k = 0$ ;

步骤三: 当  $k < k_{\max}$  时, 计算

$$x_{k+1} \leftarrow x_k - f(x_k, \lambda_k)$$

和

$$x_{k+1} \leftarrow \frac{x_{k+1}}{\|x_{k+1}\|_2};$$

步骤四：计算

$$\lambda_{k+1} \leftarrow \lambda_k + g(x_{k+1}, \lambda_k);$$

步骤五：当  $\|Ax_{k+1}^{m-1} - \lambda_{k+1}D_{k+1}x_{k+1}\|_2 < \varepsilon$  时，输出  $(x_{k+1}, \lambda_{k+1})$ ；否则返回第三步；

步骤六：结束。

为了证明算法的有效性，在下一小节，将分析算法的稳定性和收敛性。

### 3. 收敛性分析

本小节，我们讨论所提出的梯度神经网络模型(2.3)的一些收敛性质，首先我们给出定理 3.1。

**定理 3.1** 方程(2.1)的每一个解  $x_*$  都是系统(2.3)的平衡点。反过来，若  $((m-1)Ax(t)^{m-2} - \lambda(t)D)$  是非奇异，那么系统(2.3)的平衡点是方程(2.1)的解。

**证明：**前一部分，显然成立。现证明第二部分。

假定  $x_*$  是系统(2.3)的平衡点，即

$$\left. \frac{dx}{dt} \right|_{x=x_*} = -\gamma_1 \left( (m-1)Ax_*^{m-2} - \lambda_*D \right) (Ax_*^{m-1} - \lambda_*Dx_*).$$

由于  $((m-1)Ax_*^{m-2} - \lambda_*D)$  是非奇异矩阵， $\gamma_1 > 0$ ，进而有  $Ax_*^{m-1} - \lambda_*Dx_* = 0$ 。因此  $Ax_*^{m-1} = \lambda_*Dx_*$ 。证毕。

假定  $(\lambda_*, x_*)$  满足  $Ax_*^{m-1} = \lambda_*Dx_*$ 。由[13]可知，平衡点  $x_*$  附近找到邻域  $\delta > 0$ ，

$$B(x_*, \delta) := \{x \in R^n \mid \|x - x_*\|_2 < \delta\}. \quad \square$$

使得  $((m-1)Ax_*^{m-2} - \lambda_*D)$  是非奇异矩阵。

**定理 3.2** 若张量  $\mathcal{A} \in S^{[m,n]}$  满足  $Ax_*^{m-1} = \lambda_*Dx_*$ 。初始向量  $x_0 \in B(x_*, \delta)$ ，则从初始向量  $x_0$  出发的  $x$  都会收敛到  $x_*$ 。

**证明：**构造一个 Lyapunov 函数

$$L(t) = \frac{1}{2} \|E(x(t), \lambda(t))\|_2^2 = \frac{1}{2} E(x(t), \lambda(t))^T E(x(t), \lambda(t)) \geq 0.$$

求  $L(t)$  关于时间  $t$  的导数得

$$\begin{aligned} \dot{L}(t) &= \frac{dL(t)}{dt} = E(x(t), \lambda(t))^T \frac{E(x(t), \lambda(t))}{dt} \\ &= \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right)^T \left( (m-1)Ax(t)^{m-2} - \lambda(t)D \right) \frac{dx}{dt} \\ &= -\gamma_1 \text{Tr} \left[ H \phi \left( \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right)^T \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right) \right) \right], \end{aligned} \quad (3.1)$$

其中矩阵  $H = \left( (m-1)Ax(t)^{m-2} - \lambda(t)D \right)^T \left( (m-1)Ax(t)^{m-2} - \lambda(t)D \right)$  是对称正定。则对任意  $x \in B(x_*, \delta)$ ， $\left( (m-1)Ax(t)^{m-2} - \lambda(t)D \right)$  是非奇异的，我们有

$$\begin{aligned} &\lambda_{\min} \text{Tr} \left( \phi \left( \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right) \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right)^T \right) \right) \\ &\leq \text{Tr} \left( H \phi \left( \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right) \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right)^T \right) \right) \\ &\leq \lambda_{\max} \text{Tr} \left( \phi \left( \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right) \left( Ax(t)^{m-1} - \lambda(t)Dx(t) \right)^T \right) \right), \end{aligned}$$

其中  $\lambda_{\min}$  和  $\lambda_{\max}$  是矩阵  $H$  的最小和最大特征值。函数  $\phi(x) = x$  是单调递增的奇函数，所以  $-\phi(x) = \phi(-x)$ ，再加上

$$\phi(x) \begin{cases} > 0, x > 0, \\ = 0, x = 0, \\ < 0, x < 0, \end{cases}$$

我们可以得到

$$x\phi(x) \begin{cases} > 0, x \neq 0, \\ = 0, x = 0. \end{cases}$$

那么

$$\text{Tr}\left(\phi\left(\left(\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)\right)\left(\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)^T\right)\right)\right) \begin{cases} = 0, \mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t) = 0, \\ > 0, \mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t) \neq 0. \end{cases}$$

参数  $\gamma_1 > 0$ ，故  $L(t)$  关于时间  $t$  的导数满足

$$\frac{dL(t)}{dt} \begin{cases} = 0, \mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t) = 0, \\ < 0, \mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t) \neq 0. \end{cases}$$

因  $L(t)$  是正定函数， $\dot{L}(t)$  是负定函数，满足 Lyapunov 稳定性条件，故误差函数  $E(x(t), \lambda(t))$  会收敛于零。换句话说，状态向量  $x(t)$  在  $x(t)_*$  处是渐近稳定的。证毕。  $\square$

**定理 3.3** 模型(2.3)的收敛速率为  $\gamma_1\beta$ ，其中  $\lambda_{\min}(H)$ 。

**证明：**根据(3.1)，有

$$\begin{aligned} \frac{dL(t)}{dt} &= -\gamma_1 \text{Tr}\left(\phi\left(\left(\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)\right)\left(\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)^T\right)\right)\right) \\ &\leq -\gamma_1\beta \text{Tr}\left(\left(\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)\right)\left(\mathcal{A}x(t)^{m-1} - \lambda(t)Dx(t)^T\right)\right) \\ &\leq -2\gamma_1\beta L(t). \end{aligned} \tag{3.2}$$

求解(3.2)得

$$L(t) \leq L(0)e^{-2\gamma_1\beta t}.$$

因此

$$\|E(x(t), \lambda(t))\|_2 \leq \|E(x(t_0), \lambda(t_0))\|_2 e^{-\gamma_1\beta t}.$$

显然，模型的收敛速率为  $\gamma_1\beta$ 。证毕。  $\square$

## 4. 数值实验

本小节，我们用一些数值例子来表明提出的梯度神经网络求解该问题的有效性和可行性。例 4.1 和例 4.2 均在 Python 上实验。

例 4.1 [14] 令张量  $\mathcal{A} \in S^{[4,2]}$ ，其具体值如下

$$\begin{aligned} a_{1111} = a_{2222} &= \frac{4}{\sqrt{3}}, a_{1112} = a_{1211} = a_{1121} = 1, \\ a_{2122} = a_{1222} &= a_{2212} = a_{2221}. \end{aligned}$$

用梯度神经网络来求解例 4.1 时, 选择误差  $\varepsilon = 10^{-6}$  和最大迭代步数  $k_{\max} = 1000$ 。同时, 为了方便, 取  $\alpha = \alpha_1 = \alpha_2$ , 初始特征值  $\lambda_0 = 0$  和特征向量  $x_0 = (1, 0)^T$ , 获得了部分数值结果见表 1。

**Table 1.** Partial results calculated by layered neural networks (1)

**表 1.** 梯度神经网络计算出的部分结果(1)

$\lambda$	$x$	CPU(s)	IT	$\alpha_1$	error
-3.1547	$(-0.7071, -0.7071)^T$	0.0117	19	1	$6.72 \times 10^{-8}$
-3.2525	$(-0.8919, -0.4520)^T$	0.0053	17	1.1	$4.62 \times 10^{-8}$
-3.2525	$(-0.8919, -0.4520)^T$	0.0076	18	1.08	$4.01 \times 10^{-8}$

例 4.2 [15] 令张量  $\mathcal{A}(t) \in S^{[3,3]}$ , 其具体值如下

$$\begin{aligned} a_{111} &= (t-20)^2 + 2, \\ a_{211} &= a_{121} = a_{112} = \cos(t-20), \\ a_{311} &= a_{131} = a_{113} = -t + 21. \end{aligned}$$

其他  $a_{ijk} = 0$ , 其中  $t \in [0, 20]$ 。

用梯度神经方法求解张量  $\mathcal{A}(t)$  在  $t = 20$  时的特征值和特征向量, 选择初始特征值  $\lambda_0 = 1$ , 初始特征向量  $x_0 = (1, 1, 1)^T$  且误差  $\varepsilon = 10^{-7}$ 。为了简便, 取  $\alpha = \alpha_1 = \alpha_2$ 。获得了部分数值结果见表 2。

**Table 2.** Partial results calculated by layered neural networks (2)

**表 2.** 梯度神经网络计算出的部分结果(2)

$\lambda$	$x$	CPU(s)	IT	$\alpha_1$	error
2	$(1, 0, 0)^T$	0.0024	10	0.9	$1.02 \times 10^{-7}$
2	$(1, 0, 0)^T$	0.0003	1	1	0
2	$(1, 0, 0)^T$	0.0024	11	1.08	$1.02 \times 10^{-7}$

## 5. 总结

在本文中, 利用梯度神经网络方法求解含有绝对值形式的张量特征值和特征向量, 数值实验表明了该方法的有效性, 不足之处有两点: 第一, 梯度神经网络方法中的参数  $\alpha$  是影响收敛时间和误差的, 选择最佳的参数  $\alpha$  是目前急需解决的问题; 第二, 新形式的张量特征值问题涉及到绝对值函数  $|x|$ , 由于  $|x|$  是非光滑函数, 根据已有研究张量绝对值方程的工作, 绝对值函数可以用光滑函数逼近。能否用光滑函数逼近  $|x|$ , 然后再用神经网络方法求解, 这也是一个值得思考的问题。

## 参考文献

- [1] Qi, L. (2005) Eigenvalues of a Real Supersymmetric Tensor. *Journal of Symbolic Computation*, **40**, 1302-1324. <https://doi.org/10.1016/j.jsc.2005.05.007>
- [2] Li, H., Du, S. and Wang, Y. (2020) An Inexact Levenberg-Marquardt Method for Tensor Eigenvalue Complementarity Problem. *Pacific Journal of Optimization*, **16**, 87-99.
- [3] Du, S., Zhang, L., Chen, C. and Qi, L. (2018) Tensor Absolute Value Equations. *Science China Mathematics*, **61**, 1695-1710. <https://doi.org/10.1007/s11425-017-9238-6>
- [4] Zhang, Y. (2006) A Set of Nonlinear Equations and Inequalities Arising in Robotics and Its Online Solution via a Primal Neural Network. *Neurocomputing*, **70**, 513-524. <https://doi.org/10.1016/j.neucom.2005.11.006>

- 
- [5] Stanimirovic, P.S., Zivkovic, I.S. and Wei, Y. (2015) Recurrent Neural Network for Computing the Drazin Inverse. *IEEE Transactions on Neural Networks and Learning Systems*, **26**, 2830-2843. <https://doi.org/10.1109/tnnls.2015.2397551>
- [6] Zhang, Y., Chen, Z. and Chen, K. (2009) Convergence Properties Analysis of Gradient Neural Network for Solving Online Linear Equations. *Acta Automatica Sinica*, **35**, 1136-1139. <https://doi.org/10.3724/sp.j.1004.2009.01136>
- [7] Xiao, L. and Zhang, Y. (2014) Solving Time-Varying Inverse Kinematics Problem of Wheeled Mobile Manipulators Using Zhang Neural Network with Exponential Convergence. *Nonlinear Dynamics*, **76**, 1543-1559. <https://doi.org/10.1007/s11071-013-1227-7>
- [8] Chen, K. (2013) Implicit Dynamic System for Online Simultaneous Linear Equations Solving. *Electronics Letters*, **49**, 101-102. <https://doi.org/10.1049/el.2012.3501>
- [9] Yi, C. and Zhang, Y. (2008) Analogue Recurrent Neural Network for Linear Algebraic Equation Solving. *Electronics Letters*, **44**, 1078-1080. <https://doi.org/10.1049/el:20081390>
- [10] Zhang, Y. and Chen, K. (2008) Global Exponential Convergence and Stability of Wang Neural Network for Solving Online Linear Equations. *Electronics Letters*, **44**, 145-146. <https://doi.org/10.1049/el:20081928>
- [11] Ding, F. and Chen, T.W. (2005) Gradient Based Iterative Algorithms for Solving a Class of Matrix Equations. *IEEE Transactions on Automatic Control*, **50**, 1216-1221. <https://doi.org/10.1109/tac.2005.852558>
- [12] Wang, X., Che, M. and Wei, Y. (2020) Neural Network Approach for Solving Nonsingular Multi-Linear Tensor Systems. *Journal of Computational and Applied Mathematics*, **368**, Article ID: 112569. <https://doi.org/10.1016/j.cam.2019.112569>
- [13] Wilkinson, J.H. (1965) *The Algebraic Eigenvalue Problem*. Clarendon.
- [14] Chang, K.C., Pearson, K.J. and Zhang, T. (2013) Some Variational Principles for Z-Eigenvalues of Nonnegative Tensors. *Linear Algebra and Its Applications*, **438**, 4166-4182. <https://doi.org/10.1016/j.laa.2013.02.013>
- [15] Mo, C., Wang, X. and Wei, Y. (2020) Time-Varying Generalized Tensor Eigenanalysis via Zhang Neural Networks. *Neurocomputing*, **407**, 465-479. <https://doi.org/10.1016/j.neucom.2020.04.115>