

基于矩阵符号函数求解代数 Riccati 方程的 ANN 方法

张 剑 刘永清 沈建京

(华南理工大学自动控制工程系·广州, 510641)

摘要: 本文基于矩阵符号函数方法, 运用神经网络技术的智能特性, 给出了一种求解连续及离散代数 Riccati 方程的 ANN (artificial neural network) 方法. 最后给出了这种方法的应用例子, 验证了该方法的有效性及其可靠性.

关键词: Hamiltonian 矩阵; Symplectic 矩阵; 矩阵符号函数; 神经网络

1 问题的提出

考虑如下的连续及离散 Riccati 方程:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0, \quad (1)$$

$$A^T P A - APB(R + B^T P B)^{-1}B^T P A + Q - P = 0. \quad (2)$$

这里, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $P \in \mathbb{R}^{n \times n}$, $Q \in \mathbb{R}^{n \times n}$ 且为半正定矩阵, $R \in \mathbb{R}^{m \times m}$ 且为正定矩阵.

设方程(1)和(2)存在唯一的对称正定解 $P \in \mathbb{R}^{n \times n}$, 由(1)我们可以直接构造方程(1)的 Hamiltonian

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \quad (3)$$

由(2)我们可以构造方程(2)Symplectic 矩阵

$$\begin{aligned} M &= \begin{bmatrix} A + BR^{-1}B^T A^{-T} Q & -BR^{-1}B^T A^{-T} \\ A^{-T} Q & A^{-T} \end{bmatrix} \\ &= \begin{bmatrix} I & BR^{-1}B^T \\ 0 & A^T \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} = L^{-1}E \in \mathbb{R}^{2n \times 2n}. \end{aligned} \quad (4)$$

引理 1^[2] 如果 M 是 Symplectic 矩阵, 则 $(M + I)^{-1}(M - I)$ 是 Hamiltonian 矩阵, 假设这里的逆矩阵存在.

由引理 1, 我们可以得到方程(2)相应的 Hamiltonian 矩阵

$$H = (M + I)^{-1}(M - I) = (E + L)^{-1}(E - L). \quad (5)$$

对于式(3)和式(5)中的矩阵 H , 存在唯一的非奇异矩阵 S , 使得下式成立.

$$S^{-1}HS = \begin{bmatrix} J_+ & 0 \\ 0 & J_- \end{bmatrix}. \quad (6)$$

其中, J_+ , J_- 分别由矩阵 H 的所有正实特征值及所有负实特征值组成的 Jordan 块.

将矩阵 S 写成分块矩阵 $\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$, 同时将(6)式展开, 通过推导可知方程(1)和(2)有相同的解式

$$P = S_{21} S_{11}^{-1}. \quad (7)$$

从而将方程(1)和(2)的求解问题转换为求解非奇异矩阵 S 的问题.

2 求解非奇异矩阵 S 的 ANN 方法

根据(6)式及矩阵符号函数的性质,我们可以得到

$$\operatorname{sgn}(H) = S \begin{bmatrix} \operatorname{sgn}(J_+) & 0 \\ 0 & \operatorname{sgn}(J_-) \end{bmatrix} S^{-1} = S \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} S^{-1}. \quad (8)$$

而 $\operatorname{sgn}(H)$ 可通过下列基于矩阵符号函数的牛顿迭代算法计算得到.即

$$\begin{cases} C_0 = H, \\ C_{k+1} = (C_k + \gamma_k^2 C_k^{-1}) / 2\gamma_k, \\ \lim_{k \rightarrow \infty} C_k = \operatorname{sgn}(H). \end{cases} \quad (9)$$

其中, $\gamma_k = |\det C_k|^{-\frac{1}{n}}$, k 是迭代次数.

可以证明^[2,3],上述迭代算法的极限存在且全局收敛.令 $C = \operatorname{sgn}(H)$,有(8)式我们可以建立如右具有双隐层的多层前向神经网络模型(MLP).

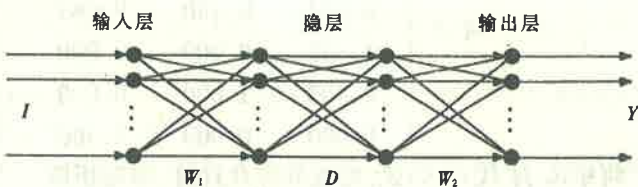


图 1 MLP 神经网络模型

这里,每层的神经元个数均为 $2n$ 个.设输入层和隐层 1 之间的权重矩阵为

W_1 ;隐层 1 和隐层 2 之间的权重矩阵为 $D = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$;隐层 2 和输出层之间的权重矩阵为 W_2 ,我们目的是通过网络训练来逼近(8)式,即

$$W_2 D W_1 \approx C = \operatorname{sgn}(H).$$

亦即要求权重矩阵 W_2 收敛于矩阵 S ; W_1 收敛于矩阵 S^{-1} ,使得 $W_2 W_1 \approx I$.采用线性函数作为神经元的激活函数,以单位矩阵的各列组成输入训练样本集 $(e_1, e_2, \dots, e_n)$,显然,对应的期望输出应为 $(Ce_1, Ce_2, \dots, Ce_n)$,从而有网络输出 $Y = W_2 D W_1 \approx C$, $S \approx W_2$.为此,我们建立如下的神经网络能量函数为

$$E = \frac{1}{2} \left[\sum_{p=1}^{2n} \sum_{q=1}^{2n} (y_{pq} - c_{pq})^2 + \sum_{p=1}^{2n} \sum_{q=1}^{2n} \left(\sum_{k=1}^{2n} w_{pk}^{(2)} w_{kq}^{(1)} - d_{pq} \right)^2 \right]. \quad (10)$$

其中, $y_{pq} = \sum_{k_1=1}^{2n} \left(\sum_{k_2=1}^{2n} w_{pk_2}^{(2)} d_{k_2 k_1} \right) w_{k_1 q}^{(1)}$, d_{pq} 为矩阵 D 的矩阵元素.

由(10)式,我们可以得到:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{(1)}} &= \sum_{r=1}^{2n} \left[(y_{rj} - c_{rj}) \sum_{k=1}^{2n} w_{rk}^{(2)} d_{ki} + \left(\sum_{k_1=1}^{2n} w_{rk_1}^{(2)} w_{k_1 j}^{(1)} - d_{rj} \right) w_{ri}^{(2)} \right], \\ \frac{\partial E}{\partial w_{ij}^{(2)}} &= \sum_{r=1}^{2n} \left[(y_{ir} - c_{ir}) \sum_{k=1}^{2n} d_{jk} w_{kr}^{(1)} + \left(\sum_{k_1=1}^{2n} w_{ik_1}^{(2)} w_{k_1 r}^{(1)} - d_{ir} \right) w_{jr}^{(1)} \right]. \end{aligned}$$

进而得到该神经网络的迭代算法为

$$w_{ij}^{(k)}(m+1) = w_{ij}^{(k)}(m) - \eta \left[\frac{\partial E}{\partial w_{ij}^{(k)}} \right](m). \quad (11)$$

其中, $k = 1, 2, 0 < \eta < 1$ 为学习因子, m 为迭代次数.通过网络训练,最终我们可以得到矩阵 $S \approx W_2$.

3 仿真例子

例 3.1 已知连续代数 Riccati 方程的系数矩阵分别为:

$$A = \begin{bmatrix} -0.459 & 0.110 & -0.291 \\ 0.321 & -0.725 & -0.310 \\ -0.044 & 0.140 & -0.683 \end{bmatrix}, \quad B = \begin{bmatrix} -0.850 & 0.000 \\ 0.000 & 1.500 \\ 1.500 & 0.000 \end{bmatrix},$$

$$Q = \begin{bmatrix} 1.500 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 1.500 \end{bmatrix}, \quad R = \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \end{bmatrix}.$$

由(3)式计算可得矩阵

$$H = \begin{bmatrix} -0.459 & 0.110 & -0.290 & -0.723 & 0.000 & 1.275 \\ 0.321 & -0.725 & -0.310 & 0.000 & -2.250 & 0.000 \\ -0.044 & 0.140 & -0.683 & 1.275 & 0.000 & -2.250 \\ -1.500 & 0.000 & 0.000 & 0.459 & -0.321 & 0.044 \\ 0.000 & -1.000 & 0.000 & -0.110 & 0.725 & -0.140 \\ 0.000 & 0.000 & -1.500 & 0.290 & 0.310 & 0.683 \end{bmatrix}.$$

将矩阵 H 代入(9)式,完成其迭代计算,得到矩阵

$$C = \operatorname{sgn}(H) = \begin{bmatrix} -0.543 & 0.093 & -0.277 & -0.482 & -0.107 & 0.584 \\ 0.261 & -0.421 & 0.034 & -0.107 & -1.339 & -0.036 \\ -0.507 & -0.042 & -0.548 & 0.584 & -0.036 & -0.903 \\ -1.759 & -0.049 & -0.535 & 0.543 & -0.261 & 0.507 \\ -0.049 & -0.592 & -0.014 & -0.093 & 0.421 & 0.042 \\ -0.535 & -0.014 & -0.969 & 0.277 & -0.039 & 0.548 \end{bmatrix}.$$

随机给定初始权重矩阵 $W_1(0)$ 和 $W_2(0)$, 根据(11)式完成网络训练可得矩阵

$$S = W_2 = \begin{bmatrix} 0.205 & -0.785 & -0.671 & 0.633 & -0.487 & 0.404 \\ 0.184 & -0.689 & -0.986 & -0.129 & 1.238 & -1.568 \\ -0.198 & 1.005 & 0.798 & 0.567 & 1.104 & -0.685 \\ -1.058 & 0.619 & 0.493 & 0.773 & -0.246 & 0.184 \\ -0.065 & 0.585 & 0.919 & 0.011 & 0.477 & -0.624 \\ -0.466 & -0.872 & -0.663 & 0.435 & 0.590 & -0.352 \end{bmatrix}.$$

由 S 矩阵可得其分块矩阵 S_{11} 和 S_{21} , 根据(7)式可得其方程的解矩阵为

$$P = \begin{bmatrix} 20.689 & -2.224 & 14.025 \\ -2.223 & 1.276 & -1.434 \\ 14.024 & -1.433 & 10.845 \end{bmatrix}.$$

离散代数 Riccati 方程的求解,应首先根据(5)式将矩阵 M 转换为矩阵 H ,而后其方程的求解过程完全同连续 Riccati 方程的求解,这里不再举例。

4 结束语

本文基于矩阵符号函数方法,结合运用神经网络技术提出了一种新的求解连续及离散代数 Riccati 方程的 ANN 方法.这种新方法的主要特点是:

1) 通过建立神经网络能量函数以及确定的网络拓扑结构,将传统的代数 Riccati 方程求解问题转换为神经网络动力系统的演化问题,开创了一种新的智能化方法。

2) 在神经网络的训练实现上, 不仅其输入输出的训练样本集容易获取, 同时, 其网络训练算法规范简明, 计算速度快而有效, 这是其他算法^[5]不可比拟的。

3) 基于这种网络训练算法可建立一相应的 RC 有源电路, 使其可通过 VLSI 硬件实现, 从而可以更加快速有效地解决大阶数以及具有实时在线要求的代数 Riccati 方程的求解问题。但是, 这里给出的网络训练算法同其他 BP 算法一样, 对某些实际应用问题, 有可能会出现其网络能量函数收敛于局部极小点, 这时, 我们可以根据实际工程中的精度要求, 适当改变网络训练时的初始权重矩阵 $W_1(0)$ 和 $W_2(0)$, 一般总能得到理想的结果。

参 考 文 献

- 1 Roberts, J. D. . Linear model reduction and solution of algebraic Riccati equation by use of the sign function. Int. J. Control, 1980, 32(4):667 - 687
- 2 Gardiner, J. D. and Laub, A. J. . A generalization of the matrix sign function solution for algebraic Riccati equation. Int. J. Control, 1986, 44(3):823 - 832
- 3 Cetin, K. Koc. . Computation of the matrix sign function using continued fraction expansion. IEEE Trans. Automat. Contr., 1994, AC-39(8):1644 - 1647
- 4 Charles, S. K. and Alan, J. L. . The matrix sign function. IEEE Trans. Automat. Contr., 1995, AC-40(8):1330 - 1348
- 5 孙增圻, 袁增任. 控制系统的计算机辅助设计. 北京: 清华大学出版社, 1988

The ANN Method for Solving Algebraic Riccati Equations Based on the Matrix Sign Function

ZHANG Jian, LIU Yongqing and SHEN Jianjing

(Department of Automatic Control Engineering, South China University of Technology, Guangzhou, 510641, PRC)

Abstract: This paper gives a artificial neural network method for solving the continuous and discrete algebraic Riccati equations based on the matrix sign function. A example verifies the effectiveness and reliableness of this method.

Key words: Hamiltonian matrix; symplectic matrix; matrix sign function; neural network

本文作者简介

张 剑 1963 年生. 1985 年毕业于装甲兵工程学院, 现为华南理工大学自动控制工程系博士生. 主要研究兴趣有神经网络理论, 并行算法, 遗传算法以及它们在动态大系统稳定性分析与镇定方面的应用.

刘永清 见本刊 1997 年第 1 期 33 页.

沈建京 1961 年生. 解放军测绘学院副教授. 主要研究兴趣有动态大系统稳定性理论, 智能稳定与镇定方法.