Gradient-based Neural Network for Online Solution of Lyapunov Matrix Equation with Li Activation Function

Shiheng Wang[1], Shidong Dai[2] and Ke Wang[2, a *]

[1]Department of Automotive Engineering, Nanyang Vocational College of Agriculture, Nanyang 473000, P.R. China

[2]Department of Mathematics, College of Sciences, Shanghai 200444, P.R. China

[a]kwang@shu.edu.cn

* corresponding author

Keywords: Gradient-based neural network, Laypunov matrix equation, Activation function

Abstract. A new type of activation function, named Li activation function, is used in gradient-based neural network (GNN) to solve Lyapunov matrix equation. With this activation function, theoretical analysis shows that GNN can converge in finite time, while it can converge only in infinite time with two conventional activation functions — linear and power-sigmoid. Computer simulation results confirm that GNN with Li activation function can not only globally converge to the solution of the Lyapunov matrix equation but also converge in finite time. GNN with the conventional two activation functions are also simulated as a contrast.

Introduction

The Lyapunov (or Lyapunov-like) matrix equations are widely used in many different engineering and scientific computing areas, such as linear algebra, control theory, boundary value problem, signal processing and optimization [1].

In recent years, due to the in-depth research in recurrent neural networks (RNN), a variety of computational methods based on neural solvers have been proposed to solve matrix equation. A quintessential example should be cited in [2], where by taking advantage of Lyapunov functional theory to ensure the asymptotic stability of uncertain fuzzy recurrent neural networks with Markovian jumping parameters a novel linear matrix inequality-based stability criterion was obtained. The Neural networks with distributed and/or time-varying delays have also been studied [3]. Subsequently, Zhang neural network (ZNN) was proposed to solve Sylvester matrix equation and matrix inversion with time-varying coefficient matrix [4, 5].

To construct a neural network, the first and foremost thing is to define a scalar-valued norm-based energy function. The minimum point (it is generally a global minimum) of the energy function corresponds to the solution of the original problem. Next one should minimize the energy function. The most common method is to find the negative gradient direction. Therefore, in [6], the gradient-based neural network (GNN) model was proposed for solving Lyapunov matrix equation. And the authors of [7] improved the GNN model by using different activation functions. However, the GNN model without activation function is equivalent to the improved GNN model with linear activation function. They also proved that the improved GNN model with power-sigmoid activation function has a superior convergence. But the improved GNN model with the suggested activation functions never converges to the accurate value in finite time.

In this paper, the improved GNN model is presented with a new activation function suggested in [8], referred to Li activation function, for solving the Lyapunov matrix equation. The global convergence and finite-time convergence are proved in theory. The upper bound of the convergent time is also given. Computer simulation results demonstrate that, by using Li activation function, the GNN model can really converge in finite time. As a comparison, GNN models with power-sigmoid and linear activation function are also simulated.

Gradient-based Neural Network (GNN) Model

Consider the Lyapunov matrix equation

$$A^{\mathrm{T}}X + XA = -C, \tag{1}$$

where $A\in R^{n\times n}$ is the coefficient matrix, and $C\in R^{n\times n}$ is positive definite. According to the traditional gradient-based algorithm, the first and foremost is to define an energy function $\varepsilon(X)$ based on a nonnegative scalar-valued norm:

$$e(X) = \frac{\|AX + XA + C\|_F^2}{2},$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm, i.e., $\|A\|_F =$, and trace($A^{\mathrm{T}}A$) is the trace of $A^{\mathrm{T}}A$. Thus, it follows

$$e(X) = \frac{\mathrm{trace}((A^{\mathrm{T}}X + XA + C)^{\mathrm{T}}(A^{\mathrm{T}}X + XA + C))}{2}.$$

With the basic differential properties of the trace of a product matrix PZQ:

$$\frac{\partial \mathrm{trace}(PZQ)}{\partial Z} = P^{\mathrm{T}}Q^{\mathrm{T}},\quad \frac{\partial \mathrm{trace}(PZ^{\mathrm{T}}Q)}{\partial Z} = QP,$$

where P, Z and Q are arbitrary matrices with appropriate order, the following can be obtained

$$\frac{\partial e(X)}{\partial X} = A(A^{\mathrm{T}}X + XA + C) + (A^{\mathrm{T}}X + XA + C)A^{\mathrm{T}}.$$

By evolving along the negative gradient of such an energy function ε(X), the following classical GNN model is taken

$$\dot{X}(t) = -\Gamma(A(A^{\mathrm{T}}X + XA + C) + (A^{\mathrm{T}}X + XA + C)A^{\mathrm{T}}),$$

where Γ is a positive definite matrix, and the time varying matrix $X(t)$, starting from an initial condition $X_0 = X(0)\in R^{n\times n}$, is the activated state matrix corresponding to the theoretical solution $X^*(t)$ of (1). Usually Γ is simply taken as γI with constant scalar $\gamma > 0$ and I is identity matrix. And γ should be set as large as the hardware permit and is generally used to scale the convergence rate [9].

In 2005, Zhang et al combined four kinds of activation functions with the ZNN model [5]. In [7], to solve the Lyapunov matrix equation the author added the four kinds of activation functions to the classical GNN model, where the author called the new GNN model the improved GNN model. Here is the improved GNN model,

$$\dot{X}(t) = -\Gamma(AF(A^{\mathrm{T}}X + XA + C) + F(A^{\mathrm{T}}X + XA + C)A^{\mathrm{T}}), \tag{2}$$

where $F(\cdot)$ is a function of matrix, defined as follows:

$$F(A) = f(a_{ij}),\ A\in R^{n\times n},\ i, j = 1, 2, \ldots, n.$$

The following is the four kinds of activation functions:

(i) linear activation function $f(x) = x$;
(ii) bipolar sigmoid activation function $f(x) = (1 - \exp(-\xi x))/(1 + \exp(-\xi x))$ with $\xi \geq 2$;
(iii) power activation function $f(x) = x^p$ with odd integer $p \geq 3$;
(iv) power-sigmoid activation function

$$f(u) = \begin{cases} x^p, & |x| \geq 1, \\ \dfrac{1+\exp(-x)}{1-\exp(-x)}\cdot\dfrac{1-\exp(-xx)}{1+\exp(-xx)}, & |x| < 1, \end{cases}$$

with suitable design parameters $\xi \geq 2$ and $p \geq 3$.

In this paper, we will combine an activation function presented in 2013 [8] and referred to Li activation function to the GNN model (2). Both theoretical analysis and numerical simulation show that when using GNN model (2) with this type of activation function to solve the Lyapunov matrix equation, the state matrix $X(t)$ can converge to the accurate solution in finite time. Here is the Li activation function,

$$f(x) = \frac{1}{2}\text{sig}^r(x) + \frac{1}{2}\text{sig}^{\frac{1}{r}}(x), \tag{3}$$

where $x\square R$, $r > 0$ is a parameter. The function $\text{sig}^r(x)$ is defined as follows

$$\text{sig}^r(x) = \begin{cases} |x|^r, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -|x|^r, & \text{if } x < 0. \end{cases} \tag{4}$$

For the Li activation function defined in (3) and (4), it's easy to see that for a positive constant ρ, the case $r = \rho$ is always same to the case $r = \frac{1}{r}$. Then it is only needed to consider the Li activation function with $0 < r \leq 1$ or $r \geq 1$. It can be seen that the Li activation function is reduced to the linear activation function when $r = 1$, and for $|x|$>>1 with r increasing, the Li activation function approaches $\frac{1}{2}\text{sig}(x)$.

Global Convergence and Finite-time Convergence

It's easy to see that both $\text{sig}^r(x)$ and $\text{sig}^{\frac{1}{r}}(x)$ are monotonically increasing odd functions. Then it can be said that the Li activation function $f(x) = \frac{1}{2}\text{sig}^r(x) + \frac{1}{2}\text{sig}^{\frac{1}{r}}(x)$ is a monotonically increasing odd function, thus the following theorem holds.

Theorem 3. *If Theorem 2 is satisfied. X^* is the unique solution of Lyapunov matrix equation* (1)*. By using the GNN model* (2) *with Li activation function, the state matrix $X(t)$ staring from any initial state X_0 always converges to X^*.*

The following theorem shows that by using the GNN model (2) with Li activation function the exact solution to the Lyapunov matrix equation can be obtained in finite time.

Theorem 4. *If Theorem 2 is satisfied. X^* is the unique solution of Lyapunov matrix equation* (1)*. By using the GNN model* (2) *with Li activation function with 0<r<1, the state matrix $X(t)$ could converge to the exact solution X^* in finite time $t < \frac{\|X_0 - X^*\|_F^2}{g(1+r)(\frac{\sqrt{2a}}{n})^{(1+r)}}$, where X_0 is the initial state matrix, a is the minimum eigenvalue of matrix $M = A^{\mathrm{T}} \oplus A^{\mathrm{T}}$ and n is the order of matrix A.*

Remark If $r > 1$, similarly, an upper bound of the convergence time can also be obtained. However, by the definition of Li activation function, $r \geq 1$ and $0 < r < 1$ is the same. So it is only need to consider $0 < r < 1$ or $r > 1$. And in either case, the convergence time t has the same supremum which is smaller than the upper bound obtained in Theorem 4.

Illustrative Example

For illustration and comparison, consider Lyapunov matrix equation (1) with the following coefficients (which is the same as Example 4.9 in [9])

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{pmatrix}.$$

Taking C as an identity matrix, if the routine $X = lyap(A^{\mathrm{T}},C)$ is used, the theoretical solution can be obtained:

$$X^* = \begin{pmatrix} 2.3 & 2.1 & 0.5 \\ 2.1 & 4.6 & 1.3 \\ 0.5 & 1.3 & 0.6 \end{pmatrix}.$$

By the global convergence (Theorem 3), the initial matrix X_0 is randomly generated within $[-2,2]_{3\times3}$. As X^* is a symmetric matrix, we only need to compute $x_{11}(t)$, $x_{12}(t)$, $x_{13}(t)$, $x_{22}(t)$, $x_{23}(t)$, $x_{33}(t)$. With Li activation function and taking r=3, γ=10, it can be seen that the neural network output $X(t)$ reaches the exact solution X^* in a period of time (approximately 6 seconds which should be close to the suprenum). And the upper bound is about 12.6 seconds by Theorem 4. This shows the finite-time convergence.

The finite-time convergence of GNN model (2) with Li activation function is compared with the GNN model (2) with the linear and power-sigmoid functions, where r=3, p=3, ξ=4, and in all the three cases γ and X_0 are all chosen as, γ=10,

$$X_0 = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{pmatrix},$$

and the error is defined as $\left\| X(t) - X^* \right\|_F$. It can seen that the GNN model (2) with Li activation function can converge to the exact solution in about 6 seconds compared to the GNN model (2) with the linear and power-sigmoid activation functions which can converge to the exact solution only in infinite time. In contrast, the GNN model (2) with power-sigmoid and linear activation functions still have a relative large error at $t = 6$. It is also seen that at the end of the simulation the GNN model (2) with power-sigmoid and linear activation functions still can't return the true solution X^*.

Comparisons of the GNN model with the Li activation function and $r = 2$, $r = 3$, $r = 4$ and $r = 5$ show that when $r > 1$ a faster convergence rate can be obtained with r increasing.

Conclusion

In this paper, a new activation function, named Li activation function, is combined with the GNN model for solving Lyapunov matrix equation. Compared with traditional activation functions such as the linear and power-sigmoid activation functions, the GNN model with Li activation function can converge to the exact solution of Lyapunov matrix equation in finite time. Also, the global convergence and finite time convergence are analyzed and proved. The upper bound of the convergence time is also given. Numerical example illustrates the global convergence and finite time convergence.

Acknowledgments

The third author was supported by National Natural Science Foundation of China (11301330) and the grant “The First-class Discipline of Universities in Shanghai”.

References

[1] T. J. Laffey, H. Sˇmigoc, Sufficient conditions on commutators for a pair of stable matrices to have a common solution to the Lyapunov equation, SIAM J. Matrix Anal. Appl. 31 (2010) 2017–2028.
[2] Y. Liu, S. Z. Li, W.Wu, R. Huang, Dynamics of a mean-shift-like algorithm and its applications on clustering, Inform. Process. Lett. 113 (2013) 8–16.
[3] R. Yang, Z. Zhang, P. Shi, Exponential stability on stochastic neural networks.with discrete interval and distributed delays, IEEE Trans. Neural,Networks 21 (2010) 169–175.
[4] Y. Zhang, D. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, IEEE Trans. Neural Networks 13 (2002) 1053–1063.
[5] Y. Zhang, S. S. Ge, Design and analysis of a general recurrent neural net work model for time-varying matrix inversion, IEEE Trans. Neural Networks 16 (2005) 1477–1490.
[6] D. Guo, C. Yi, Y. Zhang, Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving, Neurocomputing 74 (2011) 3708–3712.
[7] C. Yi, Y. Chen, Z. Lu, Improved gradient-based neural networks for online solution of Lyapunov matrix equation, Inform. Process. Lett. 111 (2011) 780–786.

[8] S. Li, S. Chen, B. Liu, Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function, Neural Process. Lett. 37 (2013) 189–205.
[9] C. Mead, M. Ismail, (Eds.), Analog VLSI Implementation of Neural Systems, Kluwer Academic, Boston, 1989.