Comparison on Gradient-Based Neural Dynamics and Zhang Neural Dynamics for Online Solution of Nonlinear Equations

Yunong Zhang, Chenfu Yi, and Weimu Ma

School of Information Science and Technology, Sun Yat-Sen University
Guangzhou 510275, China
zhynong@mail.sysu.edu.cn, ynzhang@ieee.org
http://www.ee.sysu.edu.cn/teacher/detail.asp?sn=129

Abstract. For online solution of nonlinear equation $f(x) = 0$, this paper generalizes a special kind of recurrent neural dynamics by using a recent design method proposed by Zhang *et al.* Different from gradient-based dynamics (GD), the resultant Zhang dynamics (ZD) is designed based on the elimination of an indefinite error-monitoring function (instead of the elimination of a square-based positive error-function usually associated with GD). For comparative purposes, the gradient-based dynamics is also developed and exploited for solving online such a nonlinear equation $f(x) = 0$. Computer-simulation results via power-sigmoid activation functions substantiate further the theoretical analysis and efficacy of Zhang neural dynamics on nonlinear equations solving.

Keywords: Recurrent neural networks, neural dynamics, nonlinear equations, activation functions, exponential convergence.

1 Introduction

The solution of nonlinear equations in the form of $f(x) = 0$ is widely encountered in science and engineering fields. Many computational methods and models have thus been developed. For example, numerical algorithms are employed popularly for solving such nonlinear equations [1,2,3]. However, it may not be efficient enough for most numerical algorithms because of their serial-processing nature performed on digital computers [4]. Recently, due to the in-depth research in neural networks, the dynamic-system approach using recurrent neural models has become one of the important parallel-processing methods for solving online optimization and algebraic problems [5,6,7,8,9,10,11,12,13,14,15,16,17]. Generally speaking, most reported computational-schemes are related to the gradient-descent method or other methods intrinsically designed for constant problems (or to say, time-invariant problems, static problems, or stationary problems) solving.

Different from gradient-based neural-dynamic approach [5,6,7,8,18], a special kind of recurrent neural dynamics has recently been proposed by Zhang *et al*

L. Kang et al. (Eds.): ISICA 2008, LNCS 5370, pp. 269–279, 2008.

[7,9,10] for time-varying Sylvester equation solving and time-varying matrix inversion. By following and generalizing Zhang *et al*'s design method, a neural dynamics is elegantly introduced in this paper to solve nonlinear equations. The resultant Zhang dynamics (ZD, termed and abbreviated as such for presentation convenience) is developed by defining an indefinite error-monitoring function and then making it exponentially decrease to zero. For comparative purposes, the conventional gradient-based neural dynamics is investigated as well in this paper. Theoretical and simulative results both show the effectiveness, accuracy and efficiency of the resultant Zhang neural dynamics on equations solving.

2 Problem Formulation and Neural-Dynamic Solvers

Our objective in this paper is to find solution $x \in \mathbb{R}$ so as to make the following solvable nonlinear equation hold true:

$$f(x) = 0. \tag{1}$$

For discussion and comparison purposes, x^* denotes a theoretical solution to (1). The design procedures of gradient-based neural dynamics and Zhang neural dynamics are investigated in this section as follows. In addition, some basic types of activation functions (such as linear, sigmoid, power, and power-sigmoid activation-functions) are analyzed for the convergence of the neural dynamics.

2.1 Gradient-Based Neural-Dynamics

The conventional neural-dynamic computational schemes generally belong to or relate to the gradient-descent method in optimization [5,6,7,8,18]. By such a design method, we could develop the gradient-based neural dynamics for solving the nonlinear equation $f(x) = 0$ as follows. Firstly, to solve nonlinear equation (1), the gradient-descent method requires us to define a square-based positive error-function (or termed, energy function) such as $\mathcal{E}(x) = f^2(x)$. Then, a typical continuous-time adaptation rule based on the negative-gradient information leads to the following differential equation (which we term as gradient dynamics):

$$\frac{dx(t)}{dt} = -\frac{\gamma}{2}\frac{\partial \mathcal{E}(x)}{\partial x} = -\gamma f(x)\frac{\partial f(x)}{\partial x} := -\gamma f(x) f'(x), \tag{2}$$

where γ is a positive design-parameter used to adjust the convergence rate of the neural-dynamics, and $x(t)$, starting from randomly-generated initial condition $x(0) = x_0 \in \mathbb{R}$, is the neural-activation state corresponding to theoretical solution x^*. As an extension to the above design approach and as inspired by [10], we could have the following generalized nonlinear form of gradient-based dynamics (GD) by using nonlinear activation function $\phi(\cdot)$:

$$\dot{x}(t) = -\gamma\phi\big(f(x)\big)f'(x). \tag{3}$$

In addition, it is worth mentioning that, similar to usual neural-dynamic approaches, design parameter γ in (2) and hereafter, being the reciprocal of a

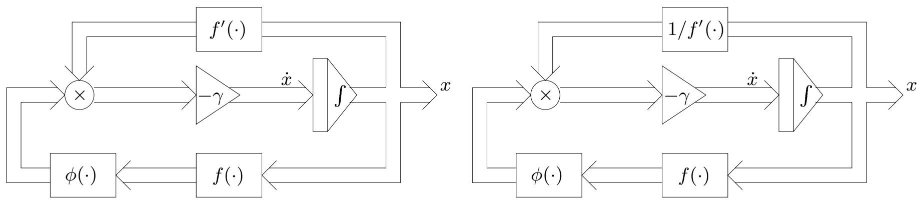


Fig. 1. Block diagrams of neural-dynamic solvers for online solution of $f(x) = 0$

capacitance parameter, could be set as large as hardware permits (e.g., in analog circuits or VLSI [7]) or selected appropriately (e.g., between 10^3 and 10^8) for experimental and/or simulative purposes.

2.2 Zhang Neural-Dynamics

By Zhang *et al*'s design method [7,9,10], we could define directly the following indefinite error function (where "indefinite" means that the function could be positive, zero, negative, and bounded or even unbounded) so as to generalize a Zhang-dynamic model for solving online the nonlinear equation $f(x) = 0$:

$$e(t) := f\big(x(t)\big).$$

Then, the time derivative $\dot{e}(t)$ of error-function $e(t)$ could be chosen and forced mathematically such that the error function $e(t)$ exponentially converges to zero. Specifically, $\dot{e}(t)$ can be chosen in the following general form:

$$\frac{de(t)}{dt} = -\gamma\phi\big(e(t)\big),$$

or equivalently,

$$\frac{df(x)}{dt} = -\gamma\phi\big(f(x)\big), \tag{4}$$

where design-parameter $\gamma > 0$ and activation-function $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$ are defined as before. Expanding the above ZD design formula, we could thus obtain

$$\frac{\partial f(x)}{\partial x}\frac{dx}{dt} = -\gamma\phi\big(f(x)\big),$$

leading to the following differential equation (which we term Zhang dynamics):

$$f'(x)\dot{x}(t) = -\gamma\phi\big(f(x)\big) \ \text{ and/or } \ \dot{x}(t) = -\gamma\phi\big(f(x)\big)/f'(x), \tag{5}$$

where $x(t)$, starting from randomly-generated initial condition $x(0) = x_0 \in \mathbb{R}$, is the neural-activation state corresponding to theoretical solution x^* of (1). In addition, it is worth pointing out that a general form of Newton-Raphson iterative method [19] for solving nonlinear equation $f(x)$=0 might be derived by discretizing this Zhang neural-dynamics instead of gradient neural-dynamics.

Moreover, the block-diagram representation of neural-dynamics (3) and (5) could be shown in Fig. 1. In view of equations (3), (5) and Fig. 1, different choices for γ and $\phi(\cdot)$ may lead to different performance of the neural-dynamics. In general, any monotonically-increasing odd activation function $\phi(\cdot)$ could be used for the construction of the neural dynamics. For example, the following four basic types of activation function $\phi(\cdot)$ could be adopted in this paper:

1) linear activation function $\phi(u) = u$,
2) bipolar sigmoid activation function (with $\xi > 2$)

$$\phi(u) = \frac{1 - \exp(-\xi u)}{1 + \exp(-\xi u)},$$

3) power activation function $\phi(u) = u^p$ with odd integer $p \geqslant 3$ (note that linear activation function $\phi(u) = u$ can be viewed as a special case of power activation function with power-index $p = 1$), and
4) power-sigmoid activation function

$$\phi(u) = \begin{cases} u^p, & \text{if } |u| \geqslant 1 \\ \frac{1+\exp(-\xi)}{1-\exp(-\xi)} \cdot \frac{1-\exp(-\xi u)}{1+\exp(-\xi u)}, & \text{otherwise} \end{cases} \tag{6}$$

with suitable design parameters $\xi \geqslant 1$ and $p \geqslant 3$.

It is worth saying that other types of activation functions can be generalized and extended by understanding the above four basic types of activation functions.

3 Theoretical Analysis

While Section 2 presents the general frameworks about gradient-based dynamics and Zhang dynamics for solving nonlinear equation $f(x) = 0$, detailed design consideration and theoretical results are given in this section. To analyze the convergence of Zhang dynamics (5), we firstly introduce the following definitions.

Definition 1. A neural-network system is said to be globally convergent, if starting from any initial state taken in the whole associated Euclidean space, every state trajectory of the neural network converges to an equilibrium point that may depend on the initial state of the system trajectory [20].

Definition 2. A neural-network system is said to be globally exponentially convergent, if every trajectory starting from any initial point $x(t_0)$ satisfies

$$\|x(t) - x^*\| \leqslant \eta \|x(t_0) - x^*\| \exp\big(-\lambda(t - t_0)\big), \ \forall t \geqslant t_0 \geqslant 0,$$

where constants $\eta > 0$ and $\lambda > 0$ exist, x^* denotes here an equilibrium point that may depend on initial state $x(t_0)$, and symbol $\|\cdot\|$ denotes the Euclidean norm for a vector (which, in our present situation, denotes the absolute value of a scalar argument) [20].

For Zhang dynamics (5), we could have the following propositions.

Proposition 1. Consider a solvable nonlinear equation $f(x) = 0$, where $f(\cdot)$ is a continuously differentiable function. If a monotonically-increasing odd activation function $\phi(\cdot)$ is employed, then neural state $x(t)$ of Zhang dynamics (5) starting from randomly-generated initial state $x(0) = x_0 \in \mathbb{R}$ could converge to theoretical solution x^* of nonlinear equation $f(x) = 0$ depicted in (1).

Proof. We can define a Lyapunov function candidate $V(x) = f^2(x)/2 \geqslant 0$, and its time derivative along the system trajectory of Zhang dynamics (5) becomes

$$\frac{dV(x)}{dt} = f(x)f'(x)\frac{dx}{dt} = -\gamma f(x)\phi\big(f(x)\big). \tag{7}$$

Because a monotonically-increasing odd function is used as an activation function, we could have $\phi\big(-f(x)\big) = -\phi\big(f(x)\big)$, and

$$\phi\big(f(x)\big) \begin{cases} > 0, & \text{if } f(x) > 0, \\ = 0, & \text{if } f(x) = 0, \\ < 0, & \text{if } f(x) < 0. \end{cases} \tag{8}$$

Hence we have

$$f(x)\phi\big(f(x)\big) \begin{cases} > 0, & \text{if } f(x) \neq 0, \\ = 0, & \text{if } f(x) = 0, \end{cases} \tag{9}$$

which guarantees the final negative-definiteness of $\dot{V}(x)$; i.e., $\dot{V}(x) < 0$ for $f(x) \neq 0$ (equivalently, $x \neq x^*$) and $\dot{V}(x) = 0$ for $f(x) = 0$ (equivalently, $x = x^*$). By Lyapunov theory [20], residual error $e(t) = f(x(t))$ could converge to zero. That is, neural state $x(t)$ of ZD (5) could converge to a theoretical solution x^* with $f(x^*) = 0$ starting from some randomly-generated initial states [note that $f'(x)$ appears in the derivation of (7)]. The proof is now complete.

Proposition 2. Let x^* denote a theoretical solution to problem $f(x) = 0$, where $f(\cdot)$ is a continuously-differentiable function (specifically, with at least first-order derivatives at some interval containing x^*). In addition to Proposition 1, neural state $x(t)$ of Zhang dynamics (5) could converge to x^* if initial state x_0 is close enough to x^*. Moreover, Zhang dynamics (5) possesses the following properties.

1) If the linear activation function is used, then exponential convergence with rate γ [in terms of residual error $e(t) = f(x(t))$] can be achieved for (5).
2) If the bipolar-sigmoid activation-function is used, superior convergence can be achieved for error range $e(t) = f(x(t)) \in [-\delta, \delta]$, $\exists \delta > 0$, as compared to the situation of using the linear activation function described in Property 1.
3) If the power activation function is used, then superior convergence can be achieved for error ranges $(-\infty, -1)$ and $(1, +\infty)$, as compared to the situation of using the linear activation function described in Property 1.

4) If the power-sigmoid activation function is used, superior convergence can be achieved for the whole error range $e(t) = f(x(t)) \in (-\infty, +\infty)$, as compared to the situation of using the linear activation function in Property 1.

Proof. We now come to prove the additional convergence properties of Zhang dynamics (5) by using the mentioned several types of activation functions $\phi(\cdot)$.

1) For the situation of using the linear activation function, it follows from equation (4) that $df(x)/dt = -\gamma f(x)$, which yields $f(x(t)) = \exp(-\gamma t) f(x_0)$. This proves the exponential convergence rate γ of Zhang dynamics (5) in the sense of its residual error $f(x(t)) \to 0$. Furthermore, we could also show the state convergence of Zhang dynamics to theoretical solution x^* via the following procedure. According to Taylor's theorem [19], we could have

$$\begin{aligned} f(x) =& f(x - x^* + x^*) \\ =& f(x^*) + (x - x^*) f'(x)|_{x=x^*} + \frac{(x-x^*)^2}{2!} f''(x)|_{x=x^*} + \cdots \\ &+ \frac{(x-x^*)^n}{n!} f^{(n)}(x)|_{x=x^*} + \frac{(x-x^*)^{n+1}}{(n+1)!} f^{(n+1)}(x)|_{x=\alpha}, \end{aligned}$$

for some α existing between $x(t)$ and x^*. Thus, in view of $f(x(t))=\exp(-\gamma t) f(x_0)$ and $f(x^*)=0$, by omitting higher-order terms, we could have $(x-x^*)f'(x)|_{x=x^*} \approx \exp(-\gamma t) f(x_0)$, which, if $f'(x)|_{x=x^*} \neq 0$, yields

$$\|x - x^*\| \approx \exp(-\gamma t) f(x_0)/f'(x)|_{x=x^*}$$

and shows the exponential convergence of Zhang dynamics (5) in terms of neural state $x(t) \to x^*$ as well. In addition, note that, even if $f^{(k)}(x)|_{x=x^*} = 0$, $\forall k = 1, 2, \cdots, (n-1)$, and $f^{(n)}(x)|_{x=x^*} \neq 0$, we still have

$$\|x - x^*\| \approx \exp(-\gamma t/n) \big(f(x_0) n! / f^n(x)|_{x=x^*} \big)^{1/n},$$

which shows again the exponential convergence of Zhang dynamics (5) in terms of neural state $x(t) \to x^*$, provided that initial state x_0 is close enough to x^*.

2) For the situation of using bipolar-sigmoid function $\phi(u) = \big(1 - \exp(-\xi u)\big)/\big(1 + \exp(-\xi u)\big)$, we know that there exists an error range $e(t) = f(x(t)) \in [-\delta, \delta]$ with $\delta > 0$ such that $\|\phi(f(x))\| > \|f(x)\|$ [7,10]. So, by reviewing the proof of Proposition 1 [especially, equation (7)], we know further that superior convergence can be achieved by ZD (5) with bipolar-sigmoid activation function for such an error range, as compared to Property 1 of using the linear function.

3) For the p-th power activation function $\phi(u) = u^p$, it follows from (4) that $df(x(t))/dt = -\gamma f^p(t)$, and its general form of trajectory is written down as

$$f(x(t)) = f(x_0) \big((p-1) f^{p-1}(x_0) \gamma t + 1 \big)^{-\frac{1}{p-1}}.$$

Specifically, for $p = 3$, residual error $f(x(t)) = f(x_0)/\sqrt{2f^2(x_0)\gamma t + 1}$. Evidently, as $t \to \infty$, $f(x) \to 0$. Look back to the proof of Proposition 1 [specifically,

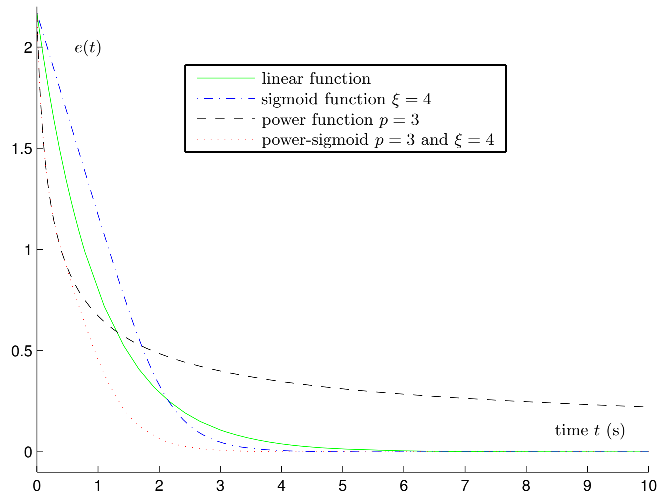


Fig. 2. Convergence behavior of error $e(t)$ with different activation functions

equation (7)]: as for Lyapunov function candidate $V(x) = f^2(x)/2$ and its time derivative $\dot{V}(x) = -\gamma f(x)\phi(f(x))$, we could have $f(x)\phi(f(x)) = f^{p+1}(x) \gg f^2(x)$ for error ranges $f(x) \ll -1$ and/or $f(x) \gg 1$. This implies that when using the power activation function, a much faster convergence can be achieved by ZD (5) for such error ranges in comparison with Property 1 [7,10].

4) It follows from the above analysis (especially, Properties 2 and 3) that, in order to achieve superior convergence, a high-performance neural dynamics could be developed by switching the power activation function to the bipolar-sigmoid activation function at switching points $f(x) = \pm 1$. Thus, if the power-sigmoid activation function is used with suitable design parameters $\xi \geqslant 1$ and $p \geqslant 3$, superior convergence could be achieved for ZD (5) theoretically over the whole error range $(-\infty, +\infty)$, as compared to the linear-activation-function situation.

For graphical interpretation, the convergence behavior of residual error $e(t)$ is illustrated in Fig. 2 by using different activation functions, where $\gamma = 1$. Note that, to draw all curves in the same one plot, small values of design parameters of nonlinear activation functions (such as $\xi = 4$ and $p = 3$) have to be used.

Remark 1. Nonlinearity always exists, which is one of the main motivations for us to investigate different activation functions in this kind of work. Even if the linear activation function is used, nonlinear phenomenon may still appear in its hardware implementation; e.g., in the form of saturation and/or inconsistency of linear slope, and in digital realization due to truncation and round-off errors [7]. The investigation of different activation functions (such as the sigmoid and power functions) may give us more insights into the imprecision problem and side-effects possibly appearing in neural hardware implementation.

Remark 2. It is worth comparing here the two design methods of neural dynamics; namely, Zhang dynamics (5) and gradient-based dynamics (3). They are both exploited for online solution of nonlinear equation (1). However, gradient-based dynamics (3) is designed based on the elimination of square-based error-function $\varepsilon(x) = f^2(x)$ as well as gradient-descent method. In contrast, Zhang dynamics (5) is designed based on the elimination of an indefinite error-function $e(t) = f(x)$ itself, which might be positive, negative, bounded or unbounded.

4 Computer-Simulation Studies

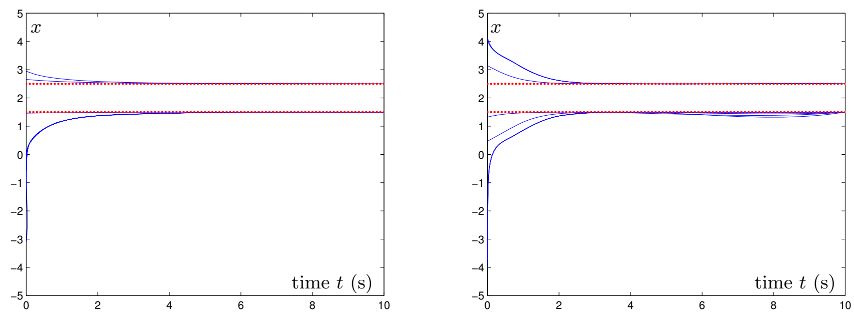
While Sections 2 and 3 present gradient-based neural dynamics (3), Zhang neural dynamics (5) and related analysis results, in this section we show computer-simulation results so as to demonstrate the characteristics of the neural-dynamic convergence. For comparative purposes, both dynamics are exploited for solving online the nonlinear equation $f(x) = 0$ with two illustrative examples below.

Example 1. Consider the following nonlinear equation in the quadratic form:

$$f(x) = x^2/2 - 2x + 1.875 = 0. \tag{10}$$

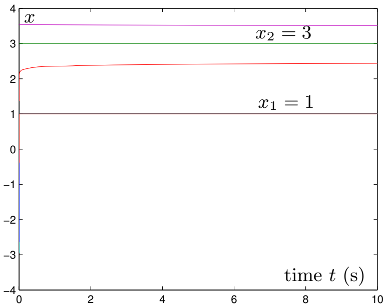
For comparative purposes, the theoretical solutions to the above nonlinear equation could be written down as $x_1^* = 1.5$ and $x_2^* = 2.5$. These are used here to verify the theoretical results discussed in the previous sections.

As seen from Fig. 3, starting from randomly-generated initial states selected within $[-5, 5]$, the activation state $x(t)$ of the investigated neural-dynamics (3) and (5) could both converge to a theoretical solution, either x_1^* or x_2^* as denoted by dotted lines in red.

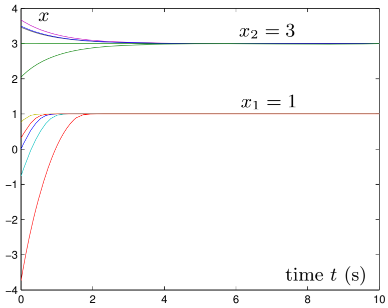


(a) Gradient dynamics (3) (b) Zhang dynamics (5)

Fig. 3. Online solution of nonlinear equation (10) by randomly-initialized GD (3) and ZD (5) with $\gamma = 1$, where dotted lines in red denote the theoretical solutions



(a) Gradient dynamics (3)



(b) Zhang dynamics (5)

Fig. 4. Online solution of (11) by randomly-initialized GD (3) and ZD (5) with $\gamma = 10$

Example 2. Consider the following nonlinear equation with multiple roots:

$$f(x) = (x-3)^{10}(x-1) = 0. \tag{11}$$

Evidently, the theoretical solutions to the above nonlinear equation could be written simply as $x_1^* = 1$ (a simple root) and $x_2^* = 3$ (a root of order 10). As seen from Fig. 4(a), by applying gradient-based neural dynamics (3), starting from randomly-generated initial states selected within $[-4, 4]$, some wrong solutions (other than $x_1^* = 1$ and $x_2^* = 3$) have also been yielded. In contrast, neural state $x(t)$ of the presented Zhang dynamics (5) could converge to a theoretical solution in this example, either $x_1^* = 1$ or $x_2^* = 3$; i.e., no wrong solution synthesized by Zhang dynamics! This could be seen clearly from Fig. 4(b).

In summary, gradient dynamics (3) could effectively solve some nonlinear equations (e.g., Example 1). However, it might yield some inapplicable solutions (e.g., Example 2) due to the multiplication of the derivative term in its dynamics. In contrast, as proved theoretically and substantiated simulatively in this paper, Zhang dynamics (5) could more accurately and effectively solve the nonlinear equation $f(x) = 0$ depicted in (1) if there exists a solution to it.

5 Conclusions

A special kind of neural dynamics has been generalized, developed, analyzed and compared in this paper by following Zhang *et al*'s method for online solution of nonlinear equations. Different from conventional gradient-based neural dynamics, such a Zhang neural dynamics has been elegantly introduced by defining an indefinite error-monitoring function (instead of usually-exploited positive energy function). Thus, its computational error could be made exponentially decrease to zero. For comparative purposes, the gradient-based neural dynamics has also been applied to online solution of such nonlinear equations. Theoretical and simulative results have both demonstrated the accuracy and efficacy of Zhang dynamic approach. Further research efforts may be directed towards the design, development and analysis of discrete-time neural models.

References

1. Yang, L., Zhu, Z., Wang, Y.: Exact Solutions of Nonlinear Equations. Physics Letters A 260, 55–59 (1999)
2. Babolian, E., Biazar, J.: Solution of Nonlinear Equations by Modified Adomian Decomposition Method. Applied Mathematics and Computation 132, 167–172 (2002)
3. Babolian, E., Biazar, J., Vahidi, A.R.: Solution of a System of Nonlinear Equations by Adomian Decomposition Method. Applied Mathematics and Computation 150, 847–854 (2004)
4. Zhang, Y., Leithead, W.E., Leith, D.J.: Time-Series Gaussian Process Regression Based on Toeplitz Computation of $O(N^2)$ Operations and $O(N)$-level Storage. In: Proceedings of IEEE Conference on Decision and Control, pp. 3711–3716 (2005)
5. Jang, J., Lee, S., Shin, S.: An Optimization Network for Matrix Inversion. In: Anderson, D.Z. (ed.) Neural Information Processing Systems, pp. 397–401. American Institute of Physics, New York (1998)
6. Wang, J.: A Recurrent Neural Network for Real-Time Matrix Inversion. Applied Mathematics and Computation 55, 89–100 (1993)
7. Zhang, Y.: Revisit the Analog Computer and Gradient-Based Neural System for Matrix Inversion. In: Proceedings of IEEE International Symposium on Intelligent Control, pp. 1411–1416 (2005)
8. Zhang, Y.: Towards Piecewise-Linear Primal Neural Networks for Optimization and Redundant Robotics. In: Proceedings of IEEE International Conference on Networking, Sensing and Control, pp. 374–379 (2006)
9. Zhang, Y., Jiang, D., Wang, J.: A Recurrent Neural Network for Solving Sylvester Equation with Time-Varying Coefficients. IEEE Transactions on Neural Networks 13, 1053–1063 (2002)
10. Zhang, Y., Ge, S.S.: Design and Analysis of a General Recurrent Neural Network Model for Time-Varying Matrix Inversion. IEEE Transactions on Neural Networks 16, 1477–1490 (2005)
11. Zhang, Y., Wang, J.: Global Exponential Stability of Recurrent Neural Networks for Synthesizing Linear Feedback Control Systems via Pole Assignment. IEEE Transactions on Neural Networks 13, 633–644 (2002)
12. Zhang, Y., Wang, J.: A Dual Neural Network for Convex Quadratic Programming Subject to Linear Equality and Inequality Constraints. Physics Letters A 298, 271–278 (2002)
13. Zhang, Y.: A Set of Nonlinear Equations and Inequalities Arising in Robotics and its Online Solution via a Primal Neural Network. Neurocomputing 70, 513–524 (2006)
14. Steriti, R.J., Fiddy, M.A.: Regularized Image Reconstruction Using SVD and a Neural Network Method for Matrix Inversion. IEEE Transactions on Signal Processing 41, 3074–3077 (1993)
15. Tank, D., Hopfield, J.: Simple Neural Optimization Networks: an A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit. IEEE Transactions on Circuits and Systems 33, 533–541 (1986)
16. Cichocki, A., Unbehauen, R.: Neural Network for Solving Systems of Linear Equations and Related Problems. IEEE Transactions on Circuits and Systems 39, 124–138 (1992)

17. Cichocki, A., Unbehauen, R.: Neural Networks for Optimization and Signal Processing. Wiley, Chichester (1993)
18. Manherz, R.K., Jordan, B.W., Hakimi, S.L.: Analog Methods for Computation of the Generalized Inverse. IEEE Transactions on Automatic Control 13, 582–585 (1968)
19. Mathews, J.H., Fink, K.D.: Numerical Methods Using MATLAB, 4th edn. Pretice Hall, New Jersey (2004)
20. Zhang, Y.: Dual Neural Networks: Design, Analysis, and Application to Redundant Robotics. In: Kang, G.B. (ed.) Progress in Neurocomputing Research, pp. 41–81. Nova Science Publishers, New York (2007)