Convergence Properties Analysis of Gradient Neural Network for Solving Online Linear Equations

ZHANG Yu-Nong[1] CHEN Zeng-Hai[1] CHEN Ke[2]

Abstract A gradient neural network (GNN) for solving online a set of simultaneous linear equations is generalized and investigated in this paper. Instead of the earlier-presented asymptotical convergence, global exponential convergence could be proved for such a class of neural networks. In addition, superior convergence could be achieved using power-sigmoid activation-functions, compared with using linear activation-functions. Computer-simulation results substantiate further the above analysis and efficacy of such neural networks.

Key words Gradient neural network (GNN), activation-function array, global exponential convergence

The efficient solution of simultaneous linear equations is viewed as a fundamental problem widely encountered in science and engineering. It is usually an essential part of many online solutions, e.g., as preliminary steps for optimization[1], signal-processing[2], electromagnetic systems[3], and robot inverse kinematics[4]. In mathematics, the problem of simultaneous linear equations could be generally formulated as

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

where coefficient matrix $A \in \mathbf{R}^{n\times n}$ and coefficient vector $\boldsymbol{b} \in \mathbf{R}^n$, while $\boldsymbol{x} \in \mathbf{R}^n$ is the unknown vector to be solved.

There are two general types of solutions to the problem of linear equations. One is the numerical algorithms performed on digital computers (i.e., on our today′s computers). Usually, the minimal arithmetic operations for a numerical algorithm are proportional to the cube of the coefficient matrix dimension n, i.e., $O(n^3)$ operations. Evidently, such serial-processing numerical algorithms may not be efficient enough for large-scale online or real-time applications. Thus, some $O(n^2)$-operation algorithms were proposed in order to remedy this computational problem[5−6]. However, they may still not be fast enough. Being the second type of solution, many parallel-processing computational methods have been developed, analyzed, and implemented on specific architectures[2,4,7−12].

As a parallel-processing computational model, a gradient neural network (GNN) was proposed by Wang[13] to solve (1) in real time. Asymptotical convergence was presented as well in [13] for such a GNN. It is worth pointing out that a pure asymptotical convergence just implies that the GNN states may only theoretically approach the solution (as time $t \to +\infty$), which may not be acceptable in practice[11]. As a result, the analysis on global exponential convergence is desirable for GNN models. In this paper, we generalize Wang′s neural network using power-sigmoid activation-functions and provide detailed analysis on global exponential convergence of the GNN models.

Received January 28, 2008; in revised form April 4, 2008
Supported by National Natural Science Foundation of China (60775050)
1. Department of Electrical and Communication Engineering, Sun-Yat Sen University, Guangzhou 510275, P. R. China 2. School of Software, Sun Yat-Sen University, Guangzhou 510275, P. R.China
DOI: 10.3724/SP.J.1004.2009.01136

The remainder of this paper is organized in four sections. Section 1 presents the problem formulation and neural-network solver. In Section 2, global exponential convergence is investigated for the neural-network solver. In Section 3, an illustrative example is given. Conclusions are finally drawn in Section 4. Before ending the section, we mention the following main contributions of this paper:

1) Global exponential convergence is proved for the GNN models solving linear equation $A\boldsymbol{x} = \boldsymbol{b}$, compared with the earlier-presented asymptotical convergence.

2) Superior convergence is proved for the generalized GNN model which exploits power-sigmoid activation-functions, compared with classic linear GNN model.

1 Problem formulation and neural solver

In order to solve (1) in parallel and in real time, a GNN[13] could be developed as the following vector-form differential equation:

$$\dot{\boldsymbol{x}}(t) = -\gamma A^{\mathrm{T}}(A\boldsymbol{x}(t) - \boldsymbol{b}) \tag{2}$$

where initial state $\boldsymbol{x}_0 = \boldsymbol{x}(0) \in \mathbf{R}^n$ and A^{T} denotes the transpose of matrix A. In addition, being the reciprocal of a capacitance parameter, design parameter $\gamma > 0$ should be implemented as large as possible. The neural-network architecture of GNN model (2) is shown in Fig. 1.

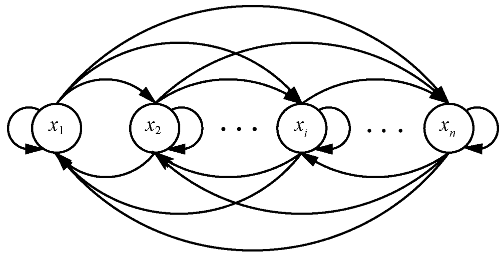


Fig. 1 Architecture and connection of GNN (2) solving $A\boldsymbol{x} = \boldsymbol{b}$

GNN (2) can be generalized using a monotonically increasing odd activation-function array:

$$\dot{\boldsymbol{x}}(t) = -\gamma A^{\mathrm{T}}\boldsymbol{F}(A\boldsymbol{x}(t) - \boldsymbol{b}) \tag{3}$$

where $\boldsymbol{F}(\cdot) : \mathbf{R}^n \to \mathbf{R}^n$ denotes an activation-function vector mapping (or an activation-function vector array) of recurrent neural networks. Note that array $\boldsymbol{F}(\cdot)$ is made of n monotonically increasing odd activation-functions $f(\cdot)$, e.g., the following two basic types depicted in Fig. 2:

1) Linear activation-function $f(e_i) = e_i$ (with e_i the i-th element of residual-error vector $\boldsymbol{e} = A\boldsymbol{x}(t) - \boldsymbol{b}$);

2) Power-sigmoid activation-function

$$f(e_i) = \begin{cases} e_i^p, & \text{if } |e_i| > 1 \\ \dfrac{1+\exp(-\xi)}{1-\exp(-\xi)} \cdot \dfrac{1-\exp(-\xi e_i)}{1+\exp(-\xi e_i)}, & \text{otherwise} \end{cases} \tag{4}$$

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

It is worth mentioning that the power-sigmoid activation-function, constructed as a combination of power and sigmoid functions, could make the neural network achieve superior convergence and robustness properties[11]. Besides, the block diagram of GNN model (3) is depicted in Fig. 3 (in addition to Fig. 1) for a better understanding of it.

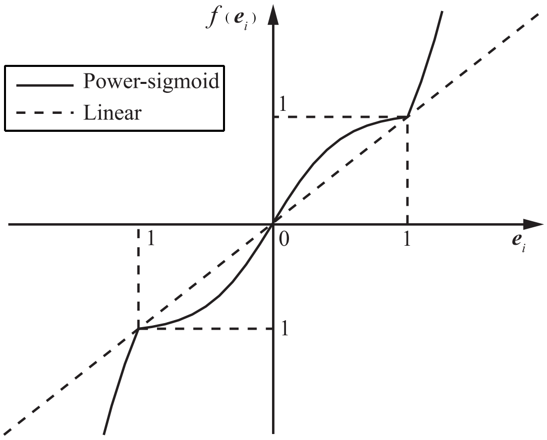


Fig. 2 Profiles of linear and nonlinear activation functions

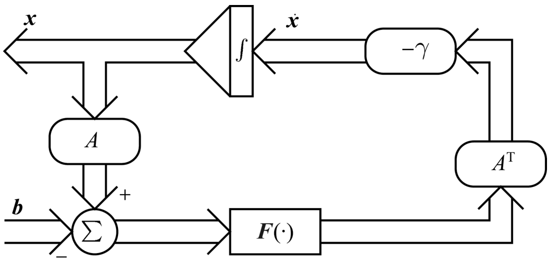


Fig. 3 Block diagram of GNN model (3) solving $A\boldsymbol{x} = \boldsymbol{b}$

2 Theoretical analysis

While Section 1 presents the GNN models (including the generalized one (3)), in this section, we will analyze their global exponential convergence and superior convergence properties of using different activation functions.

Theorem 1. Consider nonsingular constant matrix $A \in \mathbf{R}^{n\times n}$ in linear equation (1). If a linear or power-sigmoid activation-function array $\boldsymbol{F}(\cdot)$ is used, then state vector $\boldsymbol{x}(t)$ of GNN model (3) starting from any initial state $\boldsymbol{x}_0 \in \mathbf{R}^n$ will exponentially converge to the theoretical solution $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$ of linear equation $A\boldsymbol{x}(t) = \boldsymbol{b}$. In addition, the exponential-convergence rate is at least $\alpha\gamma$ with α denoting the minimum eigenvalue of $A^{\mathrm{T}}A$. Moreover, the GNN model (3) using power-sigmoid activation functions has better convergence than the GNN model using linear activation functions (equivalently, GNN (2)).

Proof. To analyze GNN models (2) and (3), let us define the solution error $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \boldsymbol{x}^*$, and let $\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$ denote the two norms of vector $\boldsymbol{x}$. Substituting $\boldsymbol{x}(t) = \tilde{\boldsymbol{x}}(t) + \boldsymbol{x}^*$ and $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$ into GNN (3), we have

$$\dot{\tilde{\boldsymbol{x}}}(t) = -\gamma A^{\mathrm{T}}\boldsymbol{F}\left(A\tilde{\boldsymbol{x}}(t)\right) \tag{5}$$

A Lyapunov function candidate can thus be defined as $v(\boldsymbol{x}) = \|\tilde{\boldsymbol{x}}(t)\|_2^2/2 = (\tilde{\boldsymbol{x}}^{\mathrm{T}}\tilde{\boldsymbol{x}})/2 \geq 0$. Evidently, $v(\boldsymbol{x})$ is positive definite in the sense that $v(\boldsymbol{x}) > 0$ for any $\tilde{\boldsymbol{x}}(t) \neq 0$ and $v(\boldsymbol{x}) = 0$ for $\tilde{\boldsymbol{x}}(t) = 0$ only (of which the latter corresponds to $\boldsymbol{x}(t) = \boldsymbol{x}^*$). In addition, $v(\boldsymbol{x}) \to \infty$ as $\|\tilde{\boldsymbol{x}}(t)\|_2 \to \infty$. Moreover, the time derivative of $v(\boldsymbol{x})$ along the system trajectory (5) is derived as

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \tilde{\boldsymbol{x}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{x}}^{\mathrm{T}}\left(-\gamma A^{\mathrm{T}}\boldsymbol{F}(A\tilde{\boldsymbol{x}})\right) = -\gamma(A\tilde{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{F}(A\tilde{\boldsymbol{x}}) \tag{6}$$

In view of processing-array $\boldsymbol{F}(\cdot)$ constituted by activation functions $f(\cdot)$ which are defined around (4) and shown in Fig. 2, we know that

$$\begin{aligned}(A\tilde{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{F}(A\tilde{\boldsymbol{x}}) = \sum_{i=1}^{n}[A\tilde{\boldsymbol{x}}]_i f\left([A\tilde{\boldsymbol{x}}]_i\right) \geq \sum_{i=1}^{n}[A\tilde{\boldsymbol{x}}]_i[A\tilde{\boldsymbol{x}}]_i = \\ \sum_{i=1}^{n}[A\tilde{\boldsymbol{x}}]_i^2 = (A\tilde{\boldsymbol{x}})^{\mathrm{T}}(A\tilde{\boldsymbol{x}}) = \|A\tilde{\boldsymbol{x}}(t)\|_2^2\end{aligned} \tag{7}$$

Then, combining (6) and (7), we have

$$\frac{\mathrm{d}v}{\mathrm{d}t} \leq -\gamma\|A\tilde{\boldsymbol{x}}\|_2^2 \leq 0 \tag{8}$$

It follows that $\dot{v}(t)$ is negative definite in the sense that $\dot{v}(t) < 0$ for any $\tilde{\boldsymbol{x}}(t) \neq 0$ and $\dot{v}(t) = 0$ for $\tilde{\boldsymbol{x}}(t) = 0$ only (corresponding to $\boldsymbol{x}(t) = \boldsymbol{x}^*$), due to matrix A being nonsingular and design-parameter $\gamma > 0$.

By Lyapunov stability theory[14−15], we have that $\tilde{\boldsymbol{x}}(t) \to 0$ as time t approaches $+\infty$; equivalently, neural state $\boldsymbol{x}(t)$ of GNN (3) is globally convergent to the theoretical inverse $\boldsymbol{x}^*$. Furthermore, by assuming $\alpha > 0$ to be the minimum eigenvalue of $A^{\mathrm{T}}A$, it follows from $\tilde{\boldsymbol{x}}^{\mathrm{T}}A^{\mathrm{T}}A\tilde{\boldsymbol{x}} \geq \alpha\tilde{\boldsymbol{x}}^{\mathrm{T}}\tilde{\boldsymbol{x}}$ and (8) that

$$\dot{v} \leq -\gamma\|A\tilde{\boldsymbol{x}}\|_2^2 = -\gamma\tilde{\boldsymbol{x}}^{\mathrm{T}}A^{\mathrm{T}}A\tilde{\boldsymbol{x}} \leq -\alpha\gamma\|\tilde{\boldsymbol{x}}\|_2^2 \leq -2\alpha\gamma v$$

Thus, $v(t) \leq \exp(-2\alpha\gamma t)v(0)$, which, together with $v(0) = \|\tilde{\boldsymbol{x}}(0)\|_2^2/2$, yields

$$\|\tilde{\boldsymbol{x}}(t)\|_2 \leq \|\tilde{\boldsymbol{x}}(0)\|_2\exp(-\alpha\gamma t), \quad t \in [0, +\infty)$$

The proof on global exponential convergence of GNN model (3) (including GNN (2)) is thus complete (with convergence rate being $\alpha\gamma$ at least).

We now come to prove the additional superior convergence of GNN model (3) using power-sigmoid activation functions compared with the situation of using linear activation functions. However, in the linear situation, since $\boldsymbol{F}(\boldsymbol{e}) = \boldsymbol{e}$ and $f(e_i) = e_i$, GNN model (3) reduces to Wang's neural network (2)[13]. Equation (5) becomes $\dot{\tilde{\boldsymbol{x}}}(t) = -\gamma A^{\mathrm{T}}(A\tilde{\boldsymbol{x}}(t))$, and the time derivative of Lyapunov function $v(\boldsymbol{x})$ becomes

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\gamma(A\tilde{\boldsymbol{x}})^{\mathrm{T}}(A\tilde{\boldsymbol{x}}) = -\gamma\|A\tilde{\boldsymbol{x}}\|_2^2 \tag{9}$$

However, in the power-sigmoid situation, let us review (5) through (8). In (7), simply put, $[A\tilde{\boldsymbol{x}}]_i f([A\tilde{\boldsymbol{x}}]_i) \geq [A\tilde{\boldsymbol{x}}]_i^2$, $\forall i \in \{1, 2, \cdots, n\}$. However, for most of the situations except $[A\tilde{\boldsymbol{x}}]_i = \pm 1 = f([A\tilde{\boldsymbol{x}}]_i)$,

$$[A\tilde{\boldsymbol{x}}]_i f([A\tilde{\boldsymbol{x}}]_i) > [A\tilde{\boldsymbol{x}}]_i^2 \text{ (for } [A\tilde{\boldsymbol{x}}]_i \in (-1, 1) \text{ or around } \pm 1)$$
$$[A\tilde{\boldsymbol{x}}]_i f([A\tilde{\boldsymbol{x}}]_i) \gg [A\tilde{\boldsymbol{x}}]_i^2 \text{ (for } |[A\tilde{\boldsymbol{x}}]_i| \gg 1)$$

In view of the above analysis and (8), we know by comparing with the linear situation (9) that the convergence speed in the power-sigmoid situation, $\dot{v}$, is at least equal to but usually faster (or much faster) than that in the linear situation (9). This means that the GNN model (3) using power-sigmoid activation functions has a superior convergence to the GNN model (3) using linear activation functions (equivalently, GNN (2)). □

Remark 1. Nonlinearity always exists, which is one of the main motivations for us to investigate power-sigmoid or other kinds of activation functions. Even if the linear activation function is preferred, the nonlinear phenomenon may appear in its hardware implementation, e.g. in the form of saturation and/or inconsistency of the linear slope and in digital realization due to truncation and round-off errors[16].

Remark 2. One more advantage of using the power-sigmoid activation function over the linear activation function lies in the extra parameters ξ and p. When there is an upper bound on γ due to hardware implementation, the new parameters ξ and p will be other effective factors expediting the neural-network convergence.

Remark 3. By following the reviewers′ inspiring and constructive comments, an approach proposed in [17], being another parallel-processing computational method for solving online linear equations, is investigated and compared as follows. 1) The approach proposed in [17], which employs a signum activation-function array, could be viewed as a special case of GNN (3). However, as the signum function is discontinuous, it may introduce extra issues of solution-existence, uniqueness, chattering, and stability analysis[18]. 2) In contrast, using linear or power-sigmoid activation-function arrays, our GNN model (3) could be proved globally stable and has no such extra solution-issues. Moreover, global exponential convergence of our GNN model (3) is also proved, being evidently more efficient and desirable.

3 Illustrative examples

For illustrative purposes, let us consider the coefficient matrix A and vector $\boldsymbol{b}$ in linear equation (1) as follows:

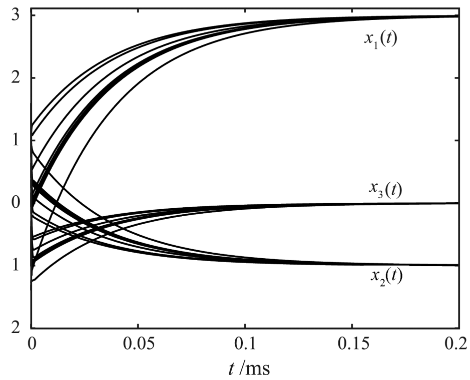
$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & -1 & -1 \\ 1 & 1 & -2 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

for which we could verify that $\boldsymbol{x}^* = [3, -1, 0]^{\mathrm{T}}$ in order to compare the correctness of the neural-network solutions.

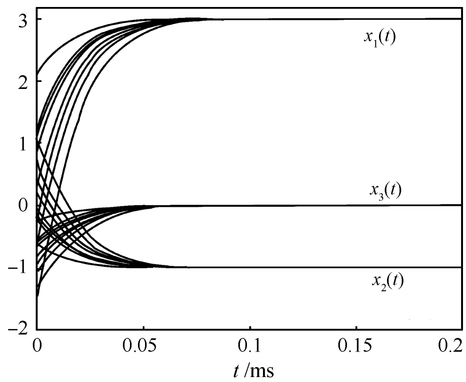
As seen from Figs. 4 $\sim$ 6, starting with any initial state randomly selected from $[-2, 2]$, state vector $\boldsymbol{x}(t)$ of GNN model (3) always converges to the theoretical solution $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$, where design parameters $\xi = 5$ and $p = 3$ are used for simulative purposes. In addition, when using power-sigmoid activation functions, GNN model (3) could converge roughly twice faster than the one using linear activation functions. Furthermore, as shown in Figs. 5 and 6, the convergence of GNN models could be expedited by increasing γ. For example, as shown in Fig. 6, the convergence time is approximately 0.1 ms when $\gamma = 10^6$, whereas the convergence time decreases to approximately 0.01 ms when γ increases to 10^7. These computer-simulation results substantiate the theoretical analysis and effectiveness of generalized GNN model (3).

4 Conclusions

Compared with the earlier-presented linear GNN model and its asymptotical convergence[13], a generalized GNN model and its global exponential convergence have been presented in this paper for linear equations solving. We show that exponential convergence can be achieved for the GNN model using linear activation functions, and that the GNN model could perform much better by using power-sigmoid activation functions than the former situation. Both theoretical analysis and simulation results have demonstrated the efficacy of GNN models on solving linear equation $A\boldsymbol{x} = \boldsymbol{b}$.

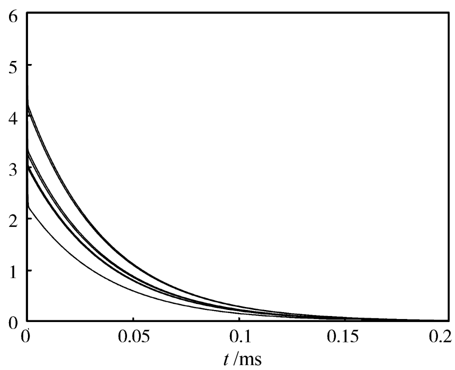


(a) Using linear activation functions

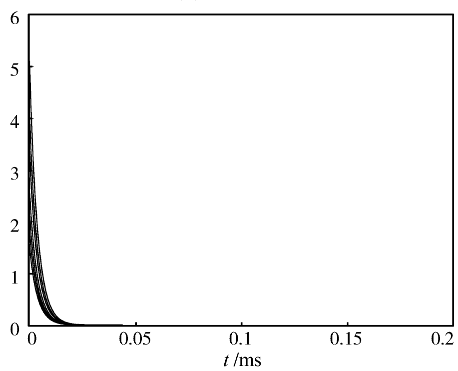


(b) Using power-sigmoid activation functions

Fig. 4 Solving $A\boldsymbol{x} = \boldsymbol{b}$ by GNN (3) using different activation functions with $\gamma = 10^6$ and starting from random initial states $\boldsymbol{x}(0)$

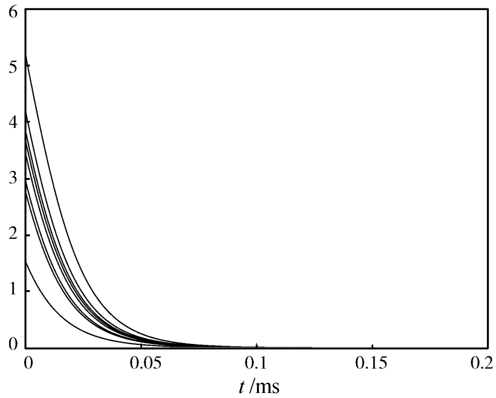


(a) $\gamma = 10^6$

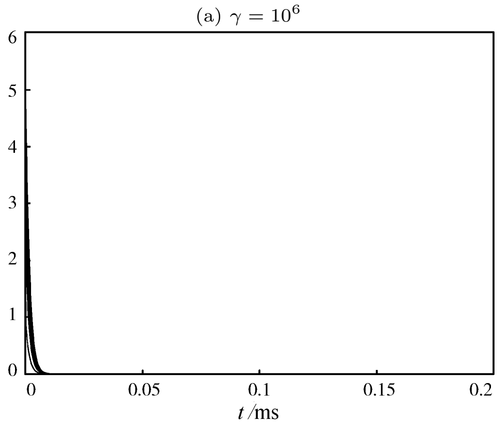


(b) $\gamma = 10^7$

Fig. 5 Solution error $\| \boldsymbol{x}(t) - \boldsymbol{x}^* \|_2$ of GNN (3) using linear activation functions



(a) $\gamma = 10^6$



(b) $\gamma = 10^7$

Fig. 6 Solution error $\| \boldsymbol{x}(t) - \boldsymbol{x}^* \|_2$ of using power-sigmoid functions

References

1 Zhang Y N. Towards piecewise-linear primal neural networks for optimization and redundant robotics. In: Proceedings of IEEE International Conference on Networking, Sensing and Control. Fort Lauderdale, USA: IEEE, 2006. 374−379

2 Steriti R J, Fiddy M A. Regularized image reconstruction using SVD and a neural network method for matrix inversion. *IEEE Transactions on Signal Processing*, 1993, **41**(10): 3074−3077

3 Sarkar T, Siarkiewicz K, Stratton R. Survey of numerical methods for solution of large systems of linear equations for electromagnetic field problems. *IEEE Transactions on Antennas Propagation*, 1981, **29**(6): 847−856

4 Sturges R H J. Analog matrix inversion (robot kinematics). *IEEE Journal of Robotics and Automation*, 1988, **4**(2): 157−162

5 Zhang Y N, 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 the 44th IEEE Conference on Decision and Control. Seville, Spain: IEEE, 2005. 3711−3716

6 Leithead W E, Zhang Y N. O(N^2)-operation approximation of covariance matrix inverse in Gaussian process regression based on quasi-Newton BFGS methods. *Communications in Statistics-Simulation and Computation*, 2007, **36**(1-3): 367−380

7 Yu You-Li, Xu Li-Hong, Wu Qi-Di. Generalized fuzzy neural network. *Acta Automatica Sinica*, 2003, **29**(6): 867−875

8 Manherz R K, Jordan B W, Hakimi S L. Analog methods for computation of the generalized inverse. *IEEE Transactions on Automatic Control*, 1968, **13**(5): 582−585

9 Zhang Y N. Revisit the analog computer and gradient-based neural system for matrix inversion. In: Proceedings of IEEE International Symposium on Intelligent Control. Cyprus, the Republic of Cyprus: IEEE, 2005. 1411−1416

10 Zhang Y N, Jiang D, Wang J. A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Transactions on Neural Networks*, 2002, **13**(5): 1053−1063

11 Zhang Y N, Ge S S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Transactions on Neural Networks*, 2005, **16**(6): 1477−1490

12 Yang Hui, Chai Tian-You. Component content soft-sensor based on neural networks in rare-earth countercurrent extraction process. *Acta Automatica Sinica*, 2006, **32**(4): 489−495

13 Wang J. Electronic realization of recurrent neural network for solving simultaneous linear equation. *Electronics Letters*, 1992, **28**(5): 493−495

14 Zhang Y N, Wang J. Global exponential stability of recurrent neural network for synthesizing linear feeback control systems via pole assignment. *IEEE Transactions on Neural Networks*, 2002, **13**(3): 633−644

15 Zhang Y N. A set of nonlinear equations and inequalities arising in robotics and its online solution via a primal neural network. *Neurocomputing*, 2006, **70**(1-3): 513−524

16 Mead C. *Analog VLSI and Neural Systems*. Massachusetts: Addison-Wesley, 1989

17 Ferreira L V, Kaszkurewicz E, Bhaya A. Solving systems of linear equations via gradient systems with discontinuous righthand sides: application to LS-SVM. *IEEE Transactions on Neural Networks*, 2005, **16**(2): 501−505

18 Bhaya A, Kaszkurewicz E. A control-theoretic approach to the design of zero finding numerical methods. *IEEE Transactions on Automatic Control*, 2007, **52**(6): 1014−1026

ZHANG Yu-Nong Professor in the Department of Electronics and Communication Engineering, Sun Yat-Sen University (SYSU). Before joining SYSU in 2006, he had been with National University of Ireland, University of Strathclyde, National University of Singapore, Chinese University of Hong Kong, from 1999. His research interest covers robotics, neural networks, and Gaussian processes. Corresponding author of this paper.
E-mails: ynzhang@ieee.org, zhynong@mail.sysu.edu.cn

CHEN Zeng-Hai Bachelor student in the Department of Electronics and Communication Engineering, Sun Yat-Sen University. His research interest covers nonlinear systems and neural networks. E-mail: czhzsu@163.com

CHEN Ke Master student at the School of Software, Sun Yat-Sen University. He received his B.S. degree in automation from Sun Yat-Sen University. His research interest covers nonlinear systems, robotics, neural networks, and signal processing.
E-mail: kechen@ieee.org