ICIC Express Letters
Volume **11**, Number **4**, April **2017**
ICIC International ©2017 ISSN 1881-803X
pp. 853–861

NEURAL NETWORK-BASED METHOD FOR SOLVING ABSOLUTE VALUE EQUATIONS

Xinjian Huang[1,2] and Baotong Cui[1,2]

[1]Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education)
[2]School of IoT Engineering
Jiangnan University
No. 1800, Lihu Ave., Wuxi 214122, P. R. China
h_xj525279@outlook.com

Received August 2016; accepted November 2016

Abstract. *This paper presents a neural network method for solving absolute value equation (AVE). First, the AVE is transformed into an equivalent linear complementary problem (LCP) when the AVE is uniquely solvable. Then, a neural network dynamic model for solving LCP is presented. By Lyapunov stability theory, the proposed neural network model is proved to be stable in the sense of Lyapunov and converges to the unique equilibrium point. Compared with existing algebraic methods based on digital computers, the proposed neural network is circuit-implementable and has a good performance in high dimensions. Moreover, the structure of the proposed neural network model is simple, which makes circuit realization easier. The simulation results show that the proposed network is feasible and efficient.*
Keywords: Absolute value equation, Linear complementary problem, Neural network, Stability

1. **Introduction.** In this paper, we consider the AVE with the form:

$$Ax - |x| = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $|x|$ denotes the components-wise absolute value of vector $x \in \mathbb{R}^n$.

The AVE arises from many mathematic problems, such as linear programs, quadratic problems, and bimatrix games. It is pointed out in [1, 2] that the AVE (1) is an NP-hard problem. In [1], Mangasarian and Meyer investigate the existence and the nonexistence of the solutions to (1). It also proved that the AVE, if uniquely solvable, can be equivalently reformulated as an LCP, which is one of the most important problems in applied science and engineering [3]. Furthermore, LCP under certain conditions can be formulated as the AVE with the form (1). This equivalent formulation has been used by Mangasarian [4] to solve the AVE using the method for solving LCP.

In recent years, the problem of finding the solutions of (1) has attracted much attention and several achievements have been obtained. A large variety of methods for solving the AVE (1) have been investigated. In [4], the AVE is reformulated as the minimization of a piecewise-linear concave function on a polyhedral set and is solved by the successive linearization method. A smoothing Newton algorithm is presented in [5], which is proved to be globally convergent with convergence rate being quadratic under the condition that the singular values of A exceed 1. Yong et al. propose an efficient harmony search algorithm for solving the AVE (1) in [6, 7]. A semismooth Newton method is proposed in [8], which largely shortens the computation time compared with the one by using the successive linearization method in [4]. Picard-Hermitian and skew-Hermitian splitting (Picard-HSS) iteration method for solving (1) is investigated in [9] which gives a sufficient condition for the convergence of the method. In [10], a relaxed nonlinear preconditioned

Hermitian and skew-Hermitian splitting-like (PHSS-like) iterative method is investigated, which is more efficient than the Picard-HSS iterative method, and is a generalization of the nonlinear Hermitian and skew-Hermitian splitting-like (HSS-like) iterative method. By using nonsmooth analysis theory, the convergence of the relaxed nonlinear PHSS-like iterative method is proved. A finite hybrid algorithm for solving the AVE (1) is presented in [11] if the AVE (1) is solvable. In [12], Iqbal et al. present a Levenberg-Marquardt method for solving (1), which can be seen as a combination of the steepest descent and the Gauss-Newton method. Using the technique of homotopy perturbation, two new iterative methods for solving the AVE are proposed in [13].

Methods mentioned above for solving the AVE (1) all rely on digital computers, which might not be efficient enough to solve the problem since the computing time required for a solution largely relies on the dimension of the equation, the complexity of the algorithm used and the performance of the computer used. Futhermore, the most obvious disadvantage of the method above is that the computing time required for a solution of (1) increases as the dimension of (1) increases; to handle these problems, one promising approach is to employ artificial neural network method based on the circuit implementation [14, 15]. The neural network is applied in a wide variety of fields including mathematical programming [16, 17], pattern recognition [18], character recognition [19], biochemistry [20], and image compression [21], etc.

The main advantage of the neural network approach is that the nature of the dynamic solution procedure is inherently parallel and distributed. Therefore, the neural network approach can solve the equation in running time at the orders of magnitude which is much faster than the general popular algorithms executed on general-purpose digital computers. In addition, neural network for solving the AVE (1) is hardware-implementable, that is to say, the neural network can be implemented by using the integrated circuits.

This paper proposes a neural network model for solving the AVE (1). First, the AVE (1) is transformed into an equivalent LCP under the condition that the singular values of A exceed 1. Then, the LCP is solved by a neural network model. The proposed method is reliable, simple and dimension independent. That is to say, the computing time required for a solution of (1) does not increase as the AVE's dimension increases and the structure of the proposed model is simpler than the model in [15]. Moreover, this proposed neural network is proved to be globally stable in the sense of Lyapunov and converges to the unique solution of the AVE.

The remainder of the paper is organized as follows. In Section 2, we describe the problem and a neural network is formulated. The stability of the proposed network is analyzed in Section 3. In Section 4, the simulation results to substantiate the theoretical arguments are given. Some conclusions are drawn in Section 5.

Throughout this paper, $\mathbb{R}^n$ denotes the space of n-dimension real column vectors, and $|x|$ denotes the components-wise absolute value of vector $x \in \mathbb{R}^n$. $\|\cdot\|$ denotes l_2 norm. I is an identity matrix with the proper dimensions, and ^{T} denotes the transpose. In what follows, eig(A) denotes the eigenvalues of A, and R^+ means the set of positive real number.

2. **AVE and Neural Network Model.**

We now show the equivalence of the AVE to a generalized LCP. Before this, the existence of the solutions to the AVE (1) is needed.

Lemma 2.1. [1]. *The AVE (1) is uniquely solvable for any $b \in \mathbb{R}^n$ if the singular values of A exceed* 1.

Remark 2.1. *In this paper, we only consider the case that the AVE (1) is uniquely solvable.*

Next, we will present that the AVE (1) is in fact equivalent to a generalized LCP.

Lemma 2.2. [1]. *The AVE (1) is equivalent to the generalized LCP*

$$((A+I)x-b)^T((A-I)x-b)=0,$$
$$(A+I)x-b\geq 0, \quad (A-I)x-b\geq 0. \tag{2}$$

Lemma 2.3. [1]. *Under the assumption that 1 is not an eigenvalue of A, the AVE (1) can be reduced to the following LCP:*

$$z^T(Mz+q)=0, \quad Mz+q\geq 0, \quad z\geq 0, \tag{3}$$

where $z=(A-I)x-b$, $M=(A+I)(A-I)^{-1}$, $q=(M-I)b$; *conversely, if* 1 *is not an eigenvalue of* M, *then the LCP (3) can be reduced to the AVE*

$$(M-I)^{-1}(M+I)x-|x|=(M-I)^{-1}q,$$

where $x=\frac{1}{2}((M-I)z+q)$.

Remark 2.2. *From Lemma 2.1 and Lemma 2.3, if the AVE (1) is uniquely solvable, then it has the same solution with the LCP (3). That is to say, to get the solution of (1), we can solve the LCP (3) instead.*

Based on the projection theorem [22], it follows that $z^*\in Z$, $Z=\{z\in\mathbb{R}^n \mid 0\leq z_i\leq +\infty, \forall i\in L\}$, $L=\{1,2,\ldots,n\}$, is a solution of the AVE defined in (3) if and only if it satisfies the following projection equation:

$$P_Z(z^*-(Mz^*+q))=z^*, \tag{4}$$

where $P_Z(z)=[P_Z(z_1),P_Z(z_2),\ldots,P_Z(z_n)]^T$ and $P_Z(z_i)=\max\{0,z_i\}$, $i=1,2,\ldots,n$.

In view of the fact that the LCP (3) is equivalent to the projection Equation (4), the following neural network is constructed for solving LCP (3):

$$\frac{dz}{dt}=\lambda(P_Z(z-(Mz+q))-z), \quad z(t_0)=z_0, \tag{5}$$

where z_0 denotes the initial points and $\lambda\in R^+$ is a scale parameter, which influences the convergence rate of the proposed neural network. An indication on how the neural network (5) can be implemented on hardware is shown in Figure 1. This neural network has n state variables, $3n$ summators and n neurons, and the structure is simple, which makes the circuit realization easier.

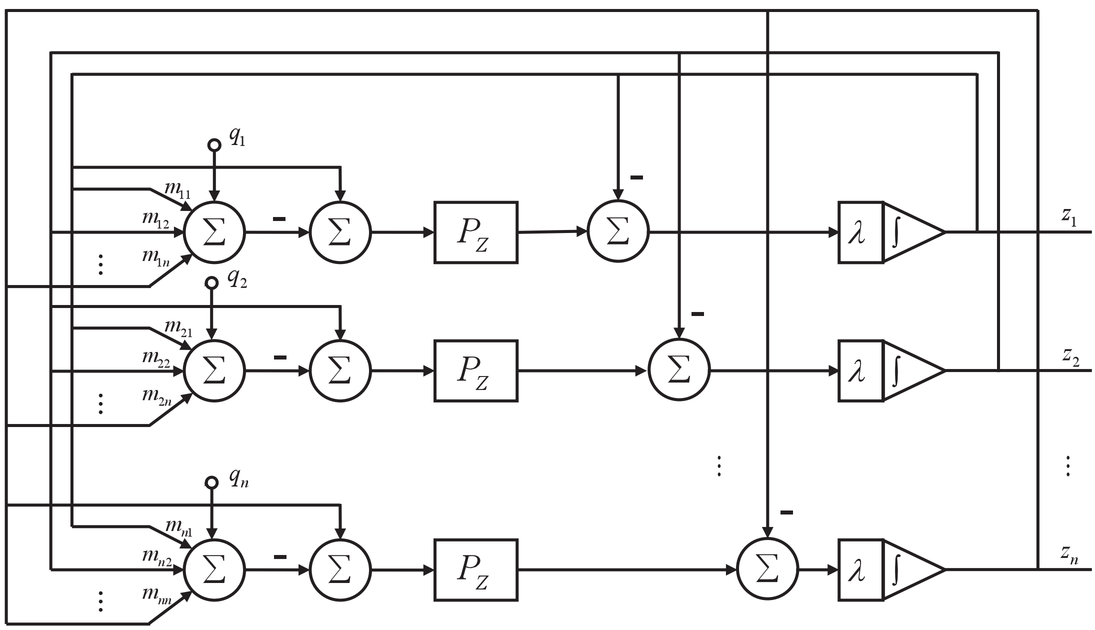


Figure 1. The architecture of the neural network (5)

In order to see how well the presented neural network (5) can be applied to solving problem (1), we compare it with the existing gradient neural network proposed in [15]. It is noticeable that the advantage of the gradient neural network model is that the neural network can be obtained directly using the derivatives of the energy function. However, its disadvantage is that the gradient neural network model only gets an approximate solution. Moreover, the circuit-implementation of the model in [15] is complex due to the calculation of the derivative of the energy function. Comparatively, the block diagram of the proposed neural network shown in Figure 1 is simple.

Remark 2.3. *Clearly, by Definition 3.1 and Lemma 2.3, the equilibrium point of the dynamic neural network model (5) is the solution of the LCP (3).*

3. **Convergence Analysis.** Before investigating the convergence properties of the neural network model (5), several definitions and lemmas are needed.

Definition 3.1. [23]. *x^* is said to be an equilibrium point of $\dot{x} = f(t, x)$ if $f(t, x^*) \equiv 0$ for all $t \geq 0$.*

Definition 3.2. [23]. *Let $D \subseteq \mathbb{R}^n$ be an open neighborhood of x^*. A continuous differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is said to be a Lyapunov function at the state x^* for a system $\dot{x} = f(t, x)$ if*

$$V(x^*) = 0, \quad V(x) > 0, \quad \forall x \in D \backslash \{x^*\},$$
$$\frac{dV(x(t))}{dt} = [\nabla V(x(t))]^T f(t, x) \leq 0, \quad \forall x \in D.$$

Lemma 3.1. [22]. *For all $v \in \mathbb{R}^n$ and all $u \in X \subset \mathbb{R}^n$*

$$(v - P_X(v))^T (P_X(v) - u) \geq 0$$

and for all $u, v \in \mathbb{R}^n$

$$\|P_X(u) - P_X(v)\| \leq \|u - v\|.$$

Now we state the main results of this section.

Theorem 3.1. *Assume that z^* is the equilibrium point of (5), and then x^* is the solution of (1).*

Proof: By Lemma 2.3 and Remark 2.3, it is easy to know that x^* is the solution of the AVE (1) if z^* is the equilibrium point of the neural network (5). This completes the proof. □

Theorem 3.2. *The proposed neural network (5) for solving the AVE (1) has a unique solution for any initial points $z_0 \in \mathbb{R}^n$.*

Proof: Let $T(z) = P_Z(z - (Mz + q)) - z$, and using Lemma 3.1 we have that for any $z_1, z_2 \in \mathbb{R}^n$

$$\begin{aligned}\|T(z_1) - T(z_2)\| &= \|P_Z(z_1 - (Mz_1 + q)) - z_1 - P_Z(z_2 - (Mz_2 + q)) + z_2\| \\ &\leq \|z_1 - (Mz_1 + q) - z_2 + (Mz_2 + q)\| + \|z_1 - z_2\| \\ &\leq \|M(z_1 - z_2)\| + 2\|z_1 - z_2\|.\end{aligned}$$

Since $Mz+q$ and z are continuously differential in $\mathbb{R}^n$, they are locally Lipschiz continuous. Thus, $T(z)$ is locally Lipschiz continuous. By the existence theory of ordinary differential equations [23] we get that for any initial points z_0 there is a unique solution of (5). This completes the proof. □

Theorem 3.3. *The matrix M in the LCP (3) is positively definite if the singular values of A exceed 1.*

Proof: Since the singular values of A exceed 1, that is to say, $\text{eig}\big(A^TA\big) > 1$. It follows that $(A-I)^{-1}$ exists and $\xi^T\big(A^TA - I\big)\xi > 0$ for $\xi \neq 0$.

Then

$$\xi^TA^TA\xi + \xi^TA^T\xi - \xi^TA\xi - \xi^T\xi > 0,$$

it yields

$$\xi^T\big(A^T - I\big)(A+I)\xi > 0, \quad \text{for} \quad \xi \neq 0.$$

Letting $\xi = (A-I)^{-1}z$, $z \neq 0$, then

$$z^T\big(A^T - I\big)^{-1}\big(A^T - I\big)(A+I)(A-I)^{-1}z > 0.$$

Thus, $z^T(A+I)(A-I)^{-1}z > 0$, $z \neq 0$. It shows that the matrix $M = (A+I)(A-I)^{-1}$ is positively definite. This completes the proof. □

Theorem 3.4. *If the matrix M is positively definite, then the neural network in (5) is stable in the sense of Lyapunov and is globally convergent to the unique solution of (5). Moreover, the convergence rate of neural network (5) increases as λ increases.*

Proof: First, consider the following function:

$$V_0(z) = \frac{1}{2}\|P_Z(z - Mz - q) - z\|^2 - (Mz+q)^T(P_Z(z - Mz - q) - z).$$

In the projection inequality of Lemma 3.1, let $v = z - Mz - q$, and $u = z$. Then

$$(z - Mz - q - P_Z(z - Mz - q))^T(P_Z(z - Mz - q) - z) \geq 0.$$

i.e.,

$$-(Mz+q)^T(P_Z(z - Mz - q) - z) \geq \|P_Z(z - Mz - q) - z\|^2.$$

Thus,

$$\begin{aligned} V_0(z) &= \frac{1}{2}\|P_Z(z - Mz - q) - z\|^2 - (Mz+q)^T(P_Z(z - Mz - q) - z) \\ &\geq \frac{3}{2}\|P_Z(z - Mz - q) - z\|^2. \end{aligned} \tag{6}$$

In what follows,

$$\begin{aligned} \frac{dV_0}{dz} &= -(P_Z(z - Mz - q) - z) - M^T(P_Z(z - Mz - q) - z) + (Mz+q) \\ &= -(M+I)^T(P_Z(z - Mz - q) - z) + (Mz+q). \end{aligned}$$

Then we have

$$\begin{aligned} \frac{dV_0}{dt} &= \left(\frac{dV_0}{dz}\right)^T\frac{dz}{dt} \\ &= \lambda(-(M+I)^T(P_Z(z - Mz - q) - z) + (Mz+q))(P_Z(z - Mz - q) - z). \end{aligned}$$

Next, define a Lyapunov function with the form:

$$V(z) = \frac{1}{2}\|z - z^*\|^2 + V_0(z), \tag{7}$$

where z^* is the only one equilibrium point of (5). By (6), it is easy to verify that $V(z) \geq 0$.

In the projection inequality of Lemma 3.1, let $v = z - Mz - q$, and $u = z^*$. Then

$$(z - Mz - q - P_Z(z - Mz - q))^T(P_Z(z - Mz - q) - z^*) \geq 0.$$

i.e.,

$$(z - Mz - q - P_Z(z - Mz - q))^T(P_Z(z - Mz - q) - z + z - z^*) \geq 0.$$

It yields

$$\begin{aligned} &(Mz + q + z - z^*)^T(P_Z(z - Mz - q) - z) \\ &\leq -(Mz+q)^T(z - z^*) - \|P_Z(z - Mz - q) - z\|^2. \end{aligned}$$

Using the above inequality, we get

$$\begin{aligned}\frac{dV(z)}{dt} &= \lambda(z - z^* - (M+I)^T(P_Z(z-Mz-q)-z) + (Mz+q))(P_Z(z-Mz-q)-z) \\ &= \lambda(Mz+q+z-z^*)^T(P_Z(z-Mz-q)-z) - \lambda\,\|P_Z(z-Mz-q)-z\|^2 \\ &\quad - \lambda(P_Z(z-Mz-q)-z)^T M(P_Z(z-Mz-q)-z) \\ &\le -\lambda(P_Z(z-Mz-q)-z)^T M(P_Z(z-Mz-q)-z) \le 0. \end{aligned} \tag{8}$$

Since M is positively definite, $-(P_Z(z-Mz-q)-z)^T M(P_Z(z-Mz-q)-z) = 0$ only if $P_Z(z-Mz-q)-z=0$. So $\frac{dV(z)}{dt} < 0$ except for $z = z^*$. It follows from Lyapunov theorem that the neural network in (5) is stable in the sense of Lyapunov. From Theorem 3.2, we get that the neural network (5) is globally convergent to the unique solution of (8). Futhermore, inequality (8) implies that a larger λ leads to a better convergence rate of $V(t)$. Therefore, the convergence rate of neural network (5) increases as λ increases. This completes the proof. □

4. **A Numerical Example.** In this section we perform a numerical example to show the feasibility of the proposed neural network (5) for solving the AVE (1). All experiments are done in a PC with 4G RAM, 3.20GHz Pentium(R) Dual-Core 32-bit processor, and all codes are written in Matlab R2011a.

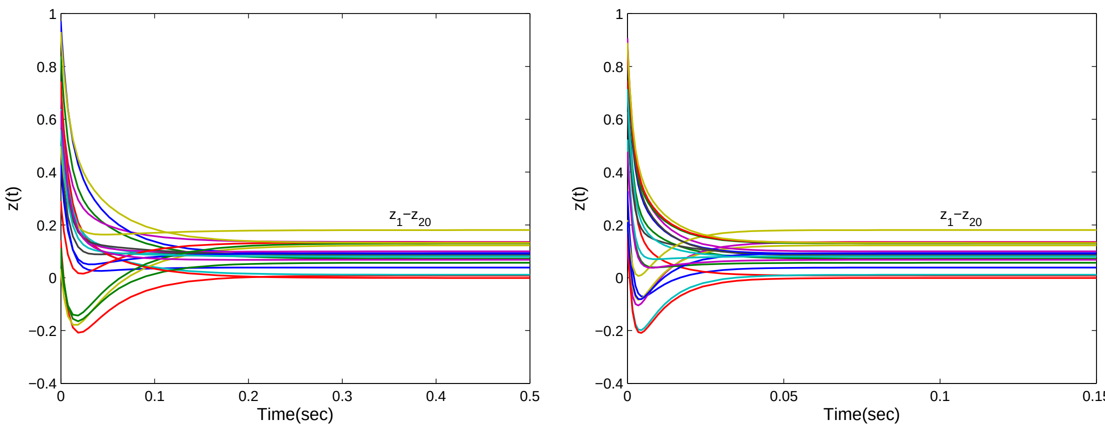
Example 4.1. *Consider a random A and b according to the following structure:*

$$A = round\left(rand(n)^T * rand(n) + n * eye(n)\right),$$

$$b = n/2 * ones(n,1),$$

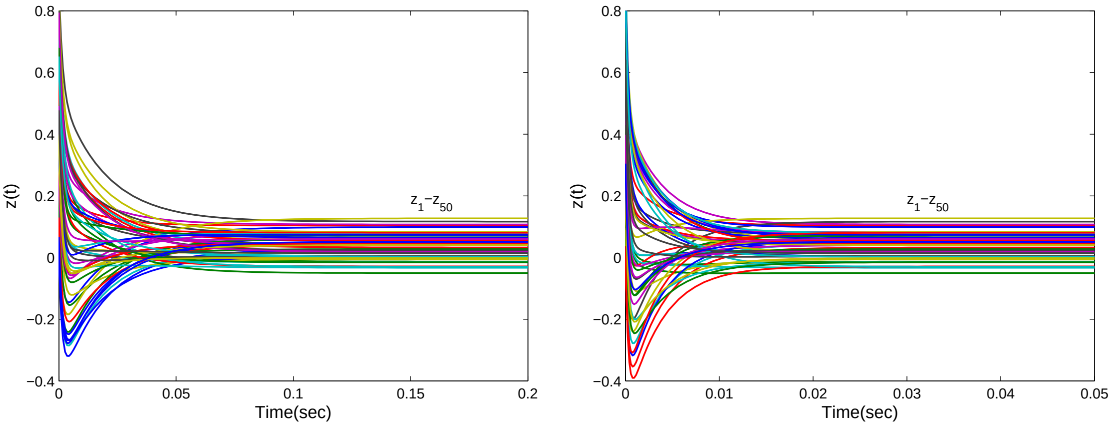
where n denotes the dimension of the considered AVE. Considering $n = 20, 50, 200, 600$ and $\lambda = 1, 5$, convergence behaviors of the z_i, $i = (1, \ldots, n)$ of neural network (5) are shown in Figures 2-5, respectively.

We can see that the neural network (5) is stable and can solve (1) efficiently. With the scale parameter λ increasing, time consumed for solving (1) decreases. Thus, by changing the value of λ, the time for solving (1) is adjustable in practical applications. Futhermore, the computing time required for a solution solved by the proposed neural network does not increase as the dimension of the AVE (1) increases. Table 1 depicts the detailed conclusion



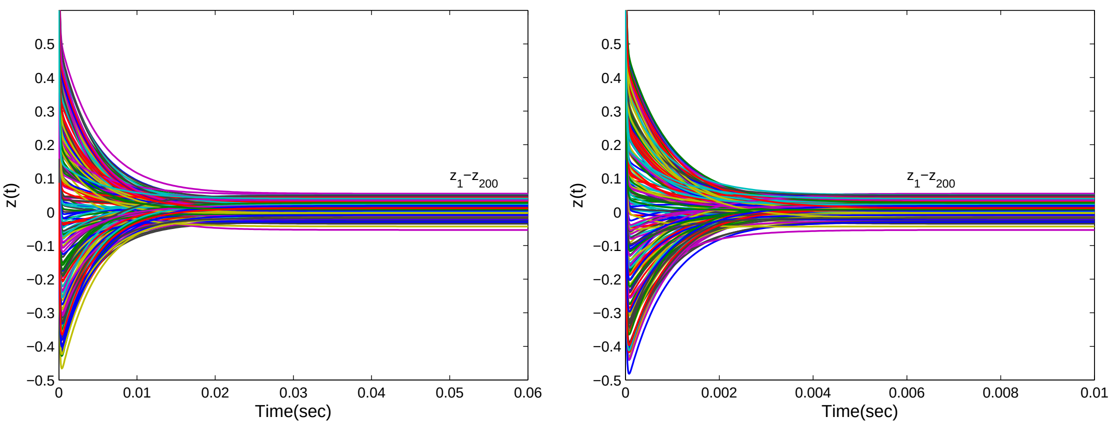
(a) $\lambda = 1$ (b) $\lambda = 5$

FIGURE 2. Transient behaviors z_i, $i = (1, \ldots, 20)$ of the neural network (5) with 20 random initial points and different λ in Example 4.1



(a) $\lambda = 1$ (b) $\lambda = 5$

FIGURE 3. Transient behaviors z_i, $i = (1, \ldots, 50)$ of the neural network (5) with 50 random initial points and different λ in Example 4.1



(a) $\lambda = 1$ (b) $\lambda = 5$

FIGURE 4. Transient behaviors z_i, $i = (1, \ldots, 200)$ of the neural network (5) with 200 random initial points and different λ in Example 4.1

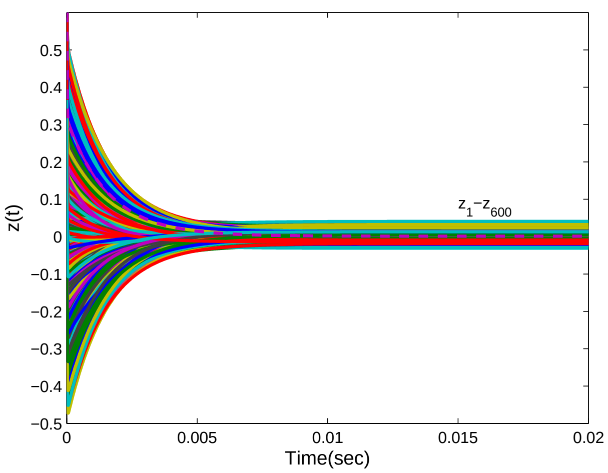


FIGURE 5. Transient behaviors z_i, $i = (1, \ldots, 600)$ of the neural network (5) with 600 random initial points and $\lambda = 1$ in Example 4.1

Table 1. Numerical results for Example 4.1

Problem size n	λ	*Error interval* ϵ	*Convergence time (s)*
20	*1*	$\leq 10^{-3}$	*less than 0.4*
20	*5*	$\leq 10^{-3}$	*less than 0.1*
50	*1*	$\leq 10^{-3}$	*less than 0.15*
50	*5*	$\leq 10^{-3}$	*less than 0.03*
200	*1*	$\leq 10^{-3}$	*less than 0.04*
200	*5*	$\leq 10^{-3}$	*less than 0.008*
600	*1*	$\leq 10^{-3}$	*less than 0.015*

of Example 4.1. In this table, the error interval $\epsilon := \max|\sigma_i|$, $i = 1, \ldots, n$, *where*

$$\sigma_i = \sum_{j=1}^{n} A_{ij} z_i^* - |z_i^*| - b_i, \quad i = 1, \ldots, n,$$

and z^* *is the equilibrium point of the corresponding neural network.*

5. **Conclusions.** In this paper, a neural network model for solving the AVE (1) has been proposed. The convergence of the proposed neural network model has been proved. Some sufficient conditions are obtained. Moreover, computing time for the solution of (1) does not increase when the dimension of (1) increases, and the error interval ϵ is adjustable in a reasonable range. Simulation results illustrate the performance of the proposed network. Considering time delay in the real-time applications and the practical circuit implementations, a delayed neural network for solving AVE (1) will be investigated in our future works.

Acknowledgment. This work is supported by National Natural Science Foundation of China (No. 61473136).

REFERENCES

[1] O. L. Mangasarian and R. Meyer, Absolute value equations, *Linear Algebra and Its Applications*, vol.419, pp.359-367, 2006.

[2] O. L. Mangasarian, Absolute value programming, *Computational Optimization and Applications*, vol.36, no.1, pp.43-53, 2007.

[3] A. Nagurney, *Variational Inequality*, Isenberg School of Management University of Massachusetts, 2002.

[4] O. L. Mangasarian, Absolute value equation solution via concave minimization, *Optimization Letters*, pp.3-8, 2007.

[5] L. Caccetta, B. Qu and G. Zhou, A globally and quadratically convergent method for absolute value equations, *Computational Optimization and Applications*, vol.48, no.1, pp.45-58, 2011.

[6] L. Yong, Y. Zhu, G. Zhang and T. Zhou, An improved harmony search algorithm for absolute value equations with normal distribution, *ICIC Express Letters*, vol.6, no.8, pp.1999-2004, 2012.

[7] L. Yong and S. Liu, An improved harmony search algorithm with differential operator for absolute value equation, *ICIC Express Letters*, vol.8, no.4, pp.1151-1157, 2014.

[8] O. L. Mangasarian, A generalized Newton method for absolute value equations, *Optimization Letters*, vol.3, pp.101-108, 2009.

[9] D. K. Salkuyeh, The Picard-HSS iteration method for absolute value equations, *Optimization Letters*, vol.8, no.8, pp.2191-2202, 2014.

[10] J. Zhang, The relaxed nonlinear PHSS-like iteration method for absolute value equations, *Applied Mathematics and Computation*, vol.265, pp.266-274, 2015.

[11] O. L. Mangasarian, A hybrid algorithm for solving the absolute value equation, *Optimization Letters*, vol.9, no.7, pp.1469-1474, 2015.

[12] J. Iqbal, A. Iqbal and M. Arif, Levenberg-Marquardt method for solving systems of absolute value equations, *Journal of Computational and Applied Mathematics*, vol.282, pp.134-138, 2015.

[13] H. Moosaei, S. Ketabchi, M. Noor, J. Iqbal and V. Hooshyarbakhsh, Some techniques for solving absolute value equations, *Applied Mathematics and Computation*, vol.268, pp.696-705, 2015.

[14] Y. Xia and J. Wang, A general methodology for designing globally convergent optimization neural networks, *IEEE Trans. Neural Networks*, vol.9, no.6, pp.1331-1343, 1998.
[15] F. R. Wang, Z. S. Yu and C. Gao, A smoothing neural network algorithm for absolute value equations, *Engineering Journal*, vol.7, no.9, pp.567-576, 2015.
[16] X. Huang, X. Lou and B. Cui, A novel neural network for solving convex quadratic programming problems subject to equality and inequality constraints, *Neurocomputing*, vol.214, pp.23-31, 2016.
[17] Y. Xia, H. Leung and J. Wang, A projection neural network and its application to constrained optimization problems, *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol.49, no.4, pp.447-458, 2002.
[18] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, 2004.
[19] C. Oprean, L. Likforman-Sulem, A. Popescu and C. Mokbel, Handwritten word recognition using Web resources and recurrent neural networks, *International Journal on Document Analysis and Recongnition*, vol.18, no.4, pp.287-301, 2015.
[20] X. Chen, A. Xue, D. Peng and Y. Guo, Modeling of pH neutralization process using fuzzy recurrent neural network and DNA based NSGA-II, *Journal of the Franklin Institute*, vol.351, pp.3847-3864, 2014.
[21] A. J. Hussain, D. Al-Jumeily, N. Radi and P. Lisboa, Hybrid neural network predictive-wavelet image compression system, *Neurocomputing*, vol.151, pp.975-984, 2015.
[22] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic, New York, 2000.
[23] K. K. Khalil, *Nonlinear Systems*, Pratice-Hall, New Jersey, 2002.