

A Smoothing Neural Network Algorithm for Absolute Value Equations

Feiran Wang, Zhensheng Yu, Chang Gao

College of Science, University of Shanghai for Science and Technology, Shanghai
Email: zhenshengyu@usst.edu.cn

Received 31 July 2015; accepted 17 September 2015; published 21 September 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we give a smoothing neural network algorithm for absolute value equations (AVE). By using smoothing function, we reformulate the AVE as a differentiable unconstrained optimization and we establish a steep descent method to solve it. We prove the stability and the equilibrium state of the neural network to be a solution of the AVE. The numerical tests show the efficiency of the proposed algorithm.

Keywords

Absolute Value Equations, Neural Network, Smoothing Function, Linear Complementarity Problem

1. Introduction

Consider the following absolute value problem [1]-[3]:

$$Ax - |x| = 0 \quad (1)$$

where $A \in R^{n \times n}$, $x, b \in R$, $|x|$ is absolute value of x , it is a subclass of absolute value equations $Ax - B|x| = b$ which is proposed by Rohn [4], and it is a NP-hard problem [1].

The AVE has closed relation with some important problems, for example, the linear programming, Quadratic programming problem and the bimatrix game problem. The above problems can be transformed into the linear complementarity problem, and the linear complementarity problem can be transformed into the absolute value equations. Due to its simple and special structure and application value, the research on absolute value equation has drawn attention of many researchers. Mangasarian [5] pointed out the relationship between backpack feasibility problem and the AVE. The problem of AVE has been studied deeply by Yamashita and Fukushima [6], and the results of the research on the problem of AVE are applied to the problem of location selection, good re-

sults are obtained. The numerical solution methods of AVE, such as Newton method, quasi-Newton method, are reachable in [7]-[12].

In this paper, we present a smooth approximation function which is based on neural network method to solve the AVE. By using a smooth approximation function of $|x|$, we turn it into a differentiable unconstrained optimization problem. Furthermore, we obtain the approximate solution of the original problem based on our established unconstrained optimization problem and the neural network model. Compared with the Newton method, the neural network model needs less requirement for the hardware of compute and the iterative process is real-time.

2. The Smoothing Reformulating of AVE

The absolute value Equation (1) is equivalent to the nonlinear equations:

$$\Theta(x) = 0 \tag{2}$$

where $\Theta(x) = Ax - |x| - b$. Since it is a non smooth function, we construct a smooth function to approximate it.

Definition 1.1 Smoothing approximation function. Given a function $\Theta: R^n \rightarrow R^n$, smoothing function $\Theta_\mu: R^n \rightarrow R^n (\mu > 0)$ is called smoothing approximation function, if for any $x \in R^n$, there exists $\kappa > 0$ so that

$$\|\Theta_\mu(\cdot) - \Theta(\cdot)\| \leq \kappa\mu, \quad \forall \mu > 0$$

where κ is not dependent on the x .

In this paper, we use the aggregate function [13] to give a smooth approximation of the absolute value equation:

Let $\phi(x) = |x|$, so every component of the absolute value function can be recorded as

$$\phi(x_i) = |x_i| = \max\{x_i, -x_i\}, \quad i = 1, 2, \dots, n$$

For any $\phi(x_i)$, the definition of smoothing function is as follows

$$\phi_\mu(x_i) = \mu \ln \left(\exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right) \right), \quad i = 1, 2, \dots, n$$

So the function of absolute value $|x|$ is obtained as follows

$$\phi_\mu(x) = (\phi_\mu(x_1), \phi_\mu(x_2), \dots, \phi_\mu(x_n))^T$$

Thus the absolute value equation is transformed into the following smooth equations

$$\Theta_\mu(x) = 0 \tag{3}$$

where $\Theta_\mu(x) = Ax - \phi_\mu(x) - b$

We define the function as follows

$$\Psi_\mu(x) = \frac{1}{2} \|\Theta_\mu(x)\|^2$$

where $\Psi_\mu(x)$ is the smoothing approximation of $\Psi(x) = \frac{1}{2} \|\Theta(x)\|^2$, $\Psi_\mu(x)$ is said as the energy function of the neural network. Thus, the approximate solution of the absolute value equation is transformed to the global optimal solution of the optimization problem $\min \Psi_\mu(x)$

3. Neural Network Model for Absolute Value Equation

Consider the following unconstrained optimization problem

$$\Psi_\mu(x) = \frac{1}{2} \|\Theta_\mu(x)\|^2 \tag{4}$$

the gradient can be calculated by the following formula:

$$\nabla \Psi_\mu(x) = [\Theta'_\mu(x)]^T \Theta_\mu(x)$$

where $\phi'_\mu(x) = \text{diag} \left(\frac{\exp\left(\frac{x_i}{\mu}\right) - \exp\left(-\frac{x_i}{\mu}\right)}{\exp\left(\frac{x_i}{\mu}\right) + \exp\left(-\frac{x_i}{\mu}\right)}, i = 1, 2, \dots, n \right)$

now, we can give a neural network model for solving the absolute value equation, which is based on the steepest descent neural network model for (4).

$$\frac{dx(t)}{dt} = -\tau \nabla \Psi_\mu(x), \quad \tau > 0 \tag{5}$$

where τ is a parameter $\tau > 1$ represents that one can use a larger step size in the simulation, specific details can be referred to [14]-[16]. To simplify our analysis, we let $\tau = 1$ throughout this paper. A block diagram (Figure 1) of the neural network is shown as follows.

4. Analysis of Stability and Existence

Next, we recall some materials about first order differential equations (ODE) [17]:

$$x'(t) = H(x(t)), \quad x(t_0) = x_0 \in R \tag{6}$$

where H is a $R^n \rightarrow R^n$ mapping. We also introduce three kinds of stability that will be discussed later.

Definition 3.1 A point $x^* = x(t^*)$ is called an equilibrium point or a steady state of the dynamic system (6) if $H(x^*) = 0$. If the neighborhood $\Omega^* \in R^n$ of x^* such that $H(x^*) = 0$ and $H(x) \neq 0, \forall x \in \Omega^* \setminus \{x^*\}$, then x^* is called an isolated equilibrium point.

Lemma 3.1 Assume that $H: R^n \rightarrow R^n$ is a continuous mapping. Then, for any $t_0 \geq 0$ and $x_0 \in R^n$, there exists a local solution $x(t)$ for (6) with $t \in [t_0, \tau]$ for some $\tau > t_0$. If, in addition, H is locally Lipschitz continuous at x_0 , then the solution is unique.

Definition 3.2 (Asymptotic stability). An isolated equilibrium point x^* is said to be asymptotically stable if in addition to being Lyapunov stable, it has the property that $x(t) \rightarrow x^*$ as $t \rightarrow \infty$ for all $\|x(t_0) - x^*\| < \delta$

Definition 3.3 (Lyapunov stability). Stability in the sense of Lyapunov Let $x(t)$ be a solution for (6). An isolated equilibrium point x^* is Lyapunov stable if for any $x(t_0) = x_0$ and any $\varepsilon > 0$ there exists a $\delta > 0$ such that $\|x(t) - x^*\| < \varepsilon$ for all $t \geq t_0$ and $\|x(t_0) - x^*\| < \delta$

Definition 3.4 (Lyapunov function). Let $\Omega \subseteq R^n$ be an open neighborhood of \bar{x} . A continuously differentiable function $W: R^n \rightarrow R^n$ is said to be a Lyapunov function at the state \bar{x} over the set Ω for Equation (6) if

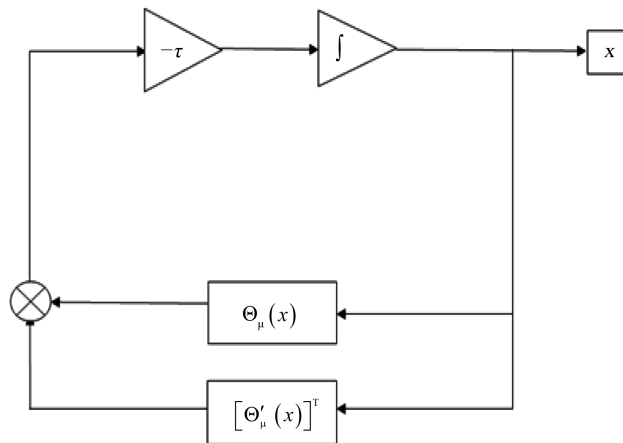


Figure 1. The block diagram of neural network (5).

$$\begin{cases} W(\bar{x}) = 0, W(x) > 0, \forall x \in \Omega \setminus \{\bar{x}\}. \\ \frac{dW(t)}{dt} = \nabla W(x(t))^T H(x(t)) \leq 0, \forall x \in \Omega. \end{cases}$$

Lemma 3.2 a) An isolated equilibrium point \bar{x} is Lyapunov stable if there exists a Lyapunov function over some neighborhood Ω^* of x^*

b) An isolated equilibrium point \bar{x} is asymptotically stable if there is a Lyapunov function over some neighborhood Ω^* of x^* such that $\frac{dW(x(t))}{dt} < 0$ for all $x \in \Omega^* \setminus \{x^*\}$

Lemma 3.3 [11] For any $\mu > 0$, $\|A^{-1}\| < 1$, if $\nabla \Psi_\mu(x) = 0$ then $\Psi(x) = 0$.

Theorem 3.1 $\Psi_\mu(x)$ is Lyapunov function over some neighborhood Ω^* of x^*

Proof. Let x^* be the solution of the absolute value equation.

1) The function $\Psi_\mu(x)$ is obtained by our smooth approximation. So $\Psi_\mu(x)$ is continuous with respect to x . Obviously $\Psi_\mu(x)$ have continuous partial derivatives at all components of the x .

2) Since $\Theta_\mu(x) = 0$, then $\Psi_\mu(x) = \frac{1}{2} \|\Theta_\mu(x)\|^2 = 0$

3) If $x \neq x^*$ then $\Psi_\mu(x) = \frac{1}{2} \|\Theta_\mu(x)\|^2 > 0$ is always holds

So, by the **Definition 3.4** we know that $\Psi_\mu(x)$ is Lyapunov function over some neighborhood Ω^* of x^*

Theorem 3.2 Each solution of the absolute value equation is the equilibrium point of the neural network (5). Conversely, if $\|A^{-1}\| < 1$, the equilibrium point of the neural network (5) is the solution of the absolute value equation.

Proof. Assume that x^* is the solution of the absolute value equation, since $\Psi_\mu(x) = \frac{1}{2} \|\Theta_\mu(x)\|^2 \geq 0$, for any $x \in R^n$ we have $\Psi(x^*) = 0$ if and only if x^* is the solution of the absolute value equation.

Obviously, we got $\nabla \Psi_\mu(x^*) = 0$, so x^* is the equilibrium point of the neural network (5). On the other hand if $\|A^{-1}\| < 1$ and $\nabla \Psi_\mu(\bar{x}) = 0$, then we get $\Psi_\mu(\bar{x}) = 0$. So, the equilibrium point of the neural network (5) is the solution of the absolute value equation.

Next, we can prove that x^* is not only Lyapunov stable and asymptotically stable.

Theorem 3.3. Let the x^* be the isolated equilibrium of the neural network. x^* is the Lyapunov stability and asymptotic stability for neural networks.

Proof. Since x^* is the isolated equilibrium of the neural network, x^* the solution of the absolute value equation is known by the **Theorem 3.2**. Therefore, $\Psi_\mu(x^*) = 0$. In addition, Since x is the isolated equilibrium point, so $\nabla \Psi_\mu(x^*) = 0$ and $\nabla \Psi_\mu(x) \neq 0, \forall x \in \Omega^* \setminus \{x^*\}$ are hold over the neighborhood Ω^* of x^* . By **Theorem 3.1** we know that $\Psi_\mu(x)$ is Lyapunov function over some neighborhood Ω^* of x^* , so by **Lemma 3.2** the isolated equilibrium x^* is Lyapunov stable. Because x^* is isolated, it is not difficult to compute:

$$\frac{d\Psi_\mu(x(t))}{dt} = \nabla \Psi_\mu(x(t))^T \frac{dx(t)}{dt} = \nabla \Psi_\mu(x(t))^T (-\tau \nabla \Psi_\mu(x(t))) = -\tau \|\Psi_\mu(x(t))\| \leq 0$$

Consequently, we have $\frac{d\Psi_\mu(x(t))}{dt} < 0, \forall x \in \Omega^* \setminus \{x^*\}$. By **Lemma 3.2**, x^* is asymptotic stability.

5. Numerical Experiment

In this section we give some smooth of numerical tests of neural network algorithm, due to the complementarity problem can be transformed to absolute value equations, we consider the linear complementarity problem which is equivalent to the absolute value equations as test cases.

For a given matrix M and vector q , The linear complementarity problem $LCP(M, q)$ is to find a vector

$x \in R^n$ to satisfy $x \geq 0, Mx + q \geq 0, x^T(Mx + q) = 0$.

From the **Theorem 2** in the literature [11], if 1 is not the eigenvalues of the matrix M , then $LCP(M, q)$ is equivalent to the following absolute value equation:

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

where $x = (M - I)z + q$ and z is the solution of the absolute value equation.

Example 1 [11]. Consider the following linear complementary problem:

$$M = \begin{pmatrix} 1 & -4 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, q = \begin{pmatrix} -5 \\ -5 \\ 1 \\ 1 \end{pmatrix}$$

Since 1 is not included in the eigenvalues of M , then the linear complementary problem can be transformed into the following absolute value equation and they are equivalent: $Ax - |x| = b$

$$\text{where } A = (M + I)(M - I)^{-1} = \begin{pmatrix} -1 & 8 & -2 & 8 \\ 0 & -1 & 0 & -2 \\ 2 & -8 & 1 & -8 \\ 0 & 2 & 0 & 1 \end{pmatrix}$$

$$b = ((M + I)(M - I)^{-1} - I)q = \begin{pmatrix} -24 \\ 8 \\ 22 \\ -10 \end{pmatrix}$$

We can find that $x^* = (-1 \ -1 \ -8 \ -4)$ is a solution of the absolute equation.

By using the neural network model, the initial point is generated by $x_0 = \text{rand}(n,1)$, and the program is performed under the environment of MATLAB7.11.0. The following two figures (Figure 2 and Figure 3) describe how the approximate solution of example 1 and the energy function varies with time.

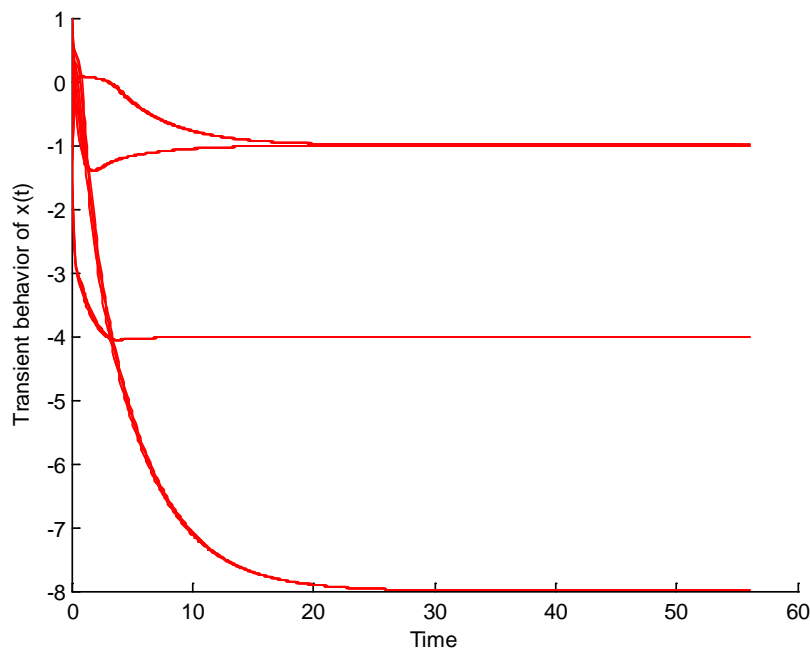


Figure 2. Transient behavior of $x(t)$ of example 1.

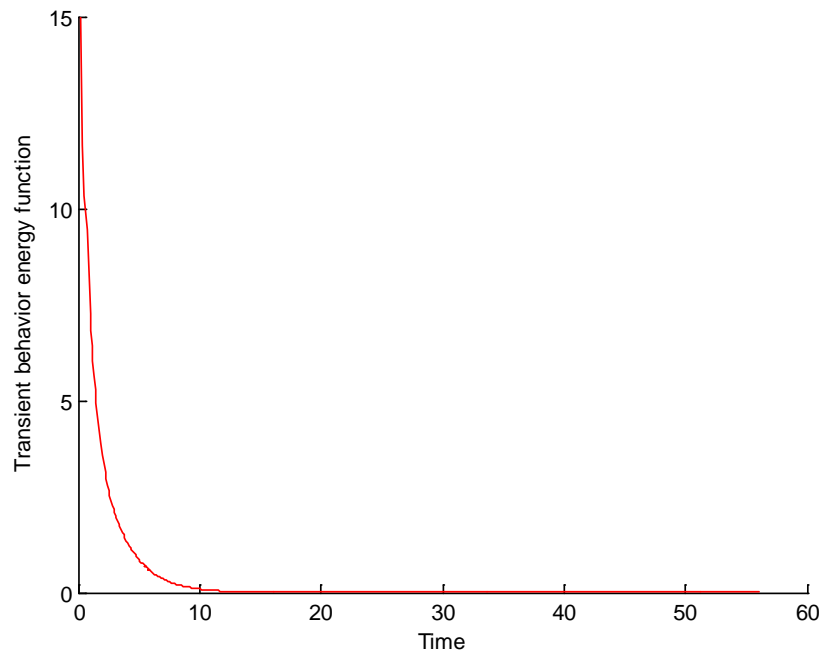


Figure 3. Transient behavior of energy function of example 1.

Since $x = (M - I)z + q$, then we can capture the solution of the linear complementary problem $LCP(M, q)$, the solution is $z^* = (1 \ 1 \ 8 \ 4)$.

Example 2 [11]. Consider the following linear complementary problem:

$$M = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} -8 \\ -6 \\ -4 \\ 3 \end{pmatrix}$$

Through calculation, we can get one eigenvalue of M is 1. By literature [11], we can find that if 1 is the eigenvalue of matrix M , then the M and q of the linear complementary problem need to be multiplied by a positive constant λ and makes 1 not the eigenvalue of M (and the solution of the linear complementary problem keeps invariant). Then we can transform linear complementary problem into absolute value equation by applied **Theorem 2** and **Theorem 3** in literature [11].

Set $\lambda = 3$, then we can find that 1 is not included in the eigenvalues of λM . And $LCP(M, q)$ and $LCP(\lambda M, \lambda q)$ have the common optimal solution, while z can be transformed into the absolute value equation by applying the **Theorem 2**. Then, we have

$$M = \begin{pmatrix} 6 & 3 & 3 & 3 \\ 3 & 6 & 0 & 3 \\ 3 & 0 & 3 & 6 \\ -3 & -3 & -6 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} -24 \\ -18 \\ -12 \\ 9 \end{pmatrix}$$

And the absolute value equation is $Ax - |x| = b$, where:

$$A = \begin{pmatrix} 2.0330 & -0.3626 & -0.2637 & 0.4286 \\ -0.3626 & 1.4890 & -0.0989 & -0.2143 \\ -0.2637 & -0.0989 & 1.1099 & -0.4286 \\ -0.4286 & 0.2143 & 0.4286 & 0.9286 \end{pmatrix}, \quad b = \begin{pmatrix} -11.2418 \\ -0.8407 \\ 2.9341 \\ 0.6429 \end{pmatrix}$$

Thus, we can get one solution of the absolute value equation which is $x^* = (-2.5 \ -0.5 \ 10.5 \ -2.5)$, then

the following two figures (Figure 4 and Figure 5) describe how the approximate solution of example 2 and the energy function varies with time

Since $x = (M - I)z + q$, then we can capture the solution of the linear complementary problem $LCP(M, q)$, the solution is $z^* = (2.5 \ 2.5 \ 0 \ 2.5)$.

Example 3. Consider the following linear complementary problem:

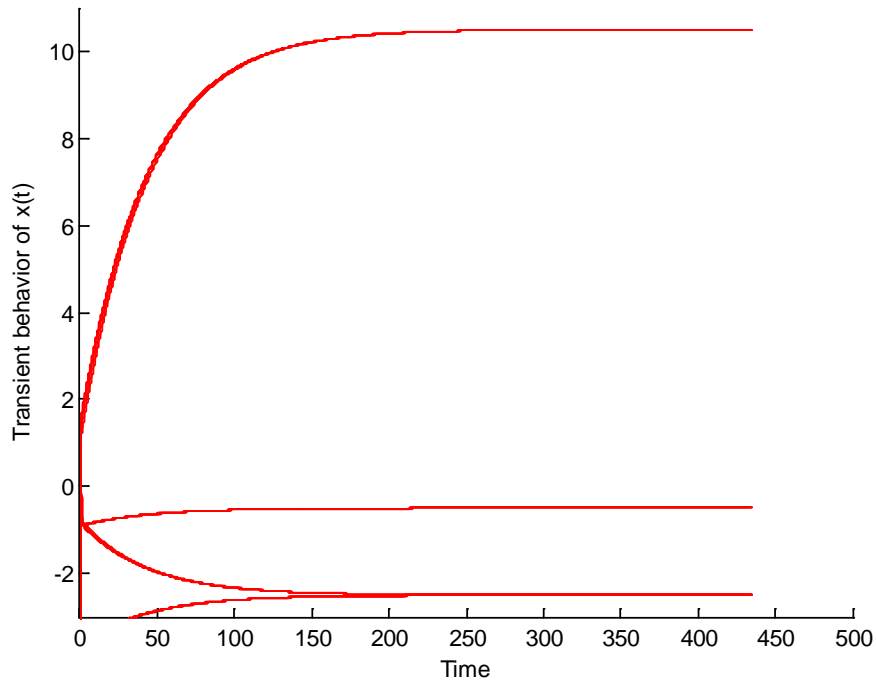


Figure 4. Transient behavior of $x(t)$ of example 2.

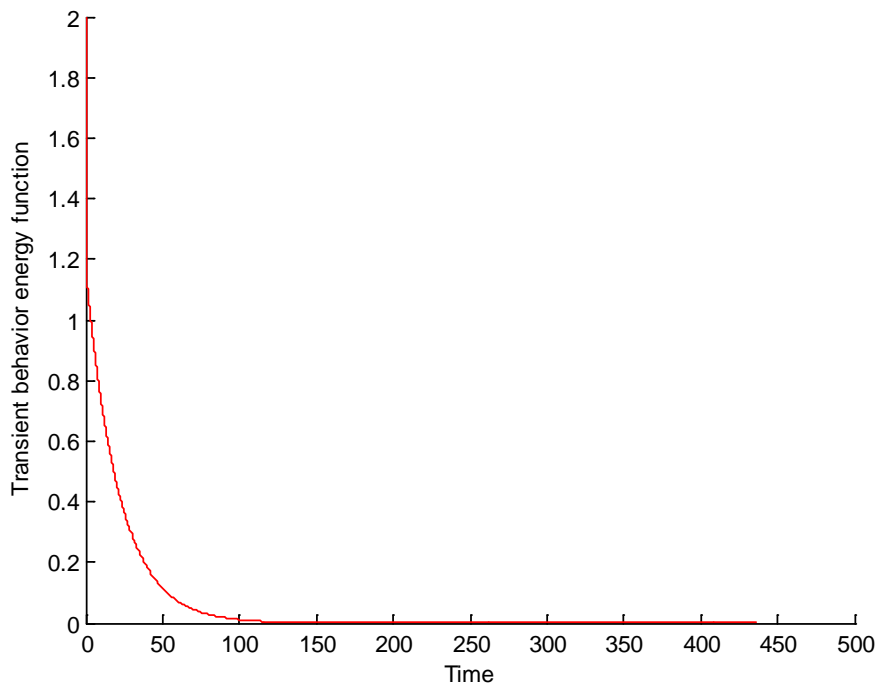


Figure 5. Transient behavior of energy function of example 2.

$$M = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, q = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$$

Through calculation we can get one eigenvalue of M is 1. And the same as example 2, set $\lambda = 3$, then, we can find that:

$$M = \begin{pmatrix} 3 & 6 & 6 & 6 \\ 0 & 3 & 6 & 6 \\ 0 & 0 & 3 & 6 \\ 0 & 0 & 0 & 3 \end{pmatrix}, q = \begin{pmatrix} -3 \\ -3 \\ -3 \\ -3 \end{pmatrix}$$

And the absolute value equation is: $Ax - |x| = b$, where:

$$A = \begin{pmatrix} 2 & -3 & 6 & -12 \\ 0 & 2 & -3 & 6 \\ 0 & 0 & 2 & -3 \\ 0 & 0 & 0 & 2 \end{pmatrix}, b = \begin{pmatrix} 24 \\ -12 \\ 6 \\ -3 \end{pmatrix}$$

Thus, we can get the solution of the absolute value equation which is $x^* = (3 \ 3 \ 3 \ -1)$, then the following two figures (Figure 6 and Figure 7) describe how the approximate solution of example 3 and the energy function varies with time

Since $x = (M - I)z + q$, then we can capture the solution of the linear complementary problem $LCP(M, q)$, the solution is $z^* = (2.5 \ 2.5 \ 0 \ 2.5)$.

6. Conclusion

This paper adopted the aggregate function method to tackle the absolute value equation with smooth processing, and then turned the absolute value equation into a differentiable unconstrained optimization problem. In order to obtain the approximate solution of the original problem we use the proposed neural network model to solve the

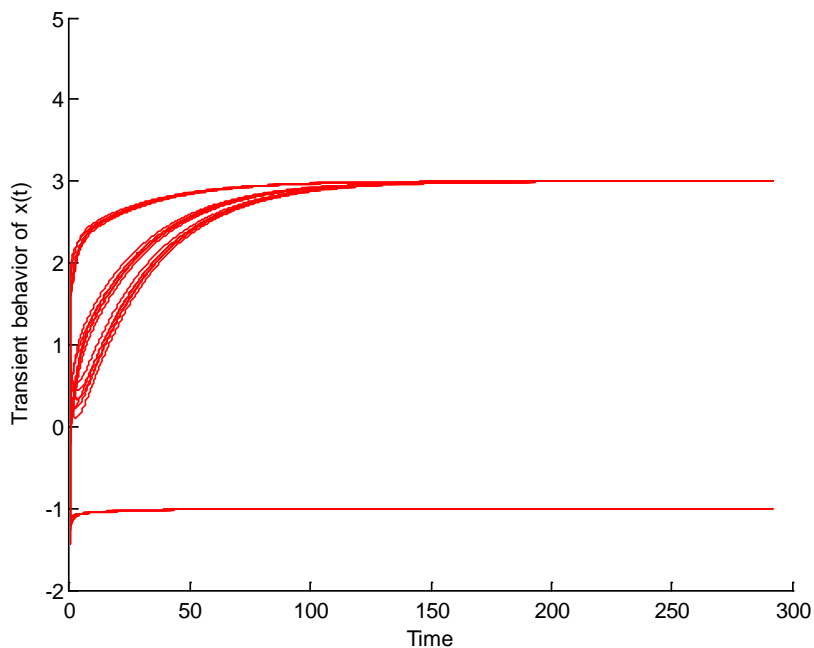


Figure 6. Transient behavior of $x(t)$ of example 3.

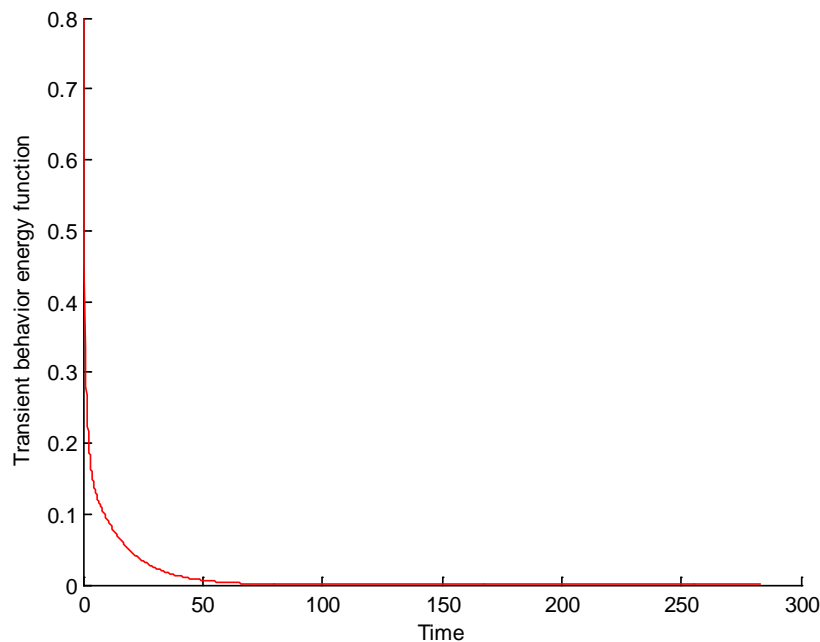


Figure 7. Transient behavior of energy function of example 3.

unconstrained optimization problem. At the same time, we propose one neural network which is based on different energy function. Through the transformation between linear complementary problem and absolute value equation, it can be used to solve the linear complementary problem, too. For the traditional energy function based on the NCP function, we can avoid a lot of matrix calculation. Numerical examples show that the algorithm is very effective for solving this kind of absolute value equation, and the accuracy of solution can be controlled by the parameters completely. In view of the fact that it is relatively difficult to solve the absolute value equation, the proposed method in this paper can be used to solve the absolute value problem effectively.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No.11171221) and Innovation Program of Shanghai Municipal Education Commission (No.14YZ094).

References

- [1] Mangasarian, O.L. and Meyer, R.R. (2006) Absolute Value Equations. *Linear Algebra and Its Applications*, **419**, 359-367. <http://dx.doi.org/10.1016/j.laa.2006.05.004>
- [2] Rohn, J. (2004) A Theorem of the Alternatives for the Equation $Ax + B|x| = b$. *Linear and Multilinear Algebra*, **52**, 421-426. <http://dx.doi.org/10.1080/0308108042000220686>
- [3] Han, J.Y., Xiu, N.H. and Qi, H.D. (2006) Nonlinear Complementary Theory and Algorithm. Shanghai Science and Technology Press, Shanghai.
- [4] Rohn, J.R. (1989) Systems of Linear Interval Equations. *Linear Algebra and Its Applications*, **126**, 39-78. [http://dx.doi.org/10.1016/0024-3795\(89\)90004-9](http://dx.doi.org/10.1016/0024-3795(89)90004-9)
- [5] Mangasarian, O.L. (2009) Knapsack Feasibility as an Absolute Value Equation Solvable by Successive Linear Programming. *Optimization Letter*, **3**, 161-170. <http://dx.doi.org/10.1007/s11590-008-0102-9>
- [6] Yamashita, S. and Fukushima, M. (2001) A Branch-and-Bound Method for Absolute Value Program and Its Application to Facility Location Problems. Kyoto University, Kyoto.
- [7] Mangasarian, O.L. (2013) Absolute Value Equation Solution via Dual Complementarity. *Optimization Letters*, **7**, 625-630. <http://dx.doi.org/10.1007/s11590-012-0469-5>
- [8] Mangasarian, O.L. (2009) A Generalized Newton Method for Absolute Value Equations. *Optimization Letters*, **3**, 101-108. <http://dx.doi.org/10.1007/s11590-008-0094-5>

- [9] Caccetta, L., Qu, B. and Zhou, G.L. (2011) A Globally and Quadratically Convergent Method for Absolute Value Equations. *Computational Optimization and Applications*, **48**, 45-58. <http://dx.doi.org/10.1007/s10589-009-9242-9>
- [10] Rohn, J. (2009) An Algorithm for Solving the Absolute Value Equation. *Electronic Journal of Linear Algebra*, **18**, 589-599.
- [11] Yong, L.Q. and Tuo, S.H. (2012) Quasi-Newton Method to Absolute Value Equations Based on Aggregate Function. *Journal of System science and Mathematics*, **32**, 1427-1436.
- [12] Yong, L.Q., Liu, S.Y. and Tuo, S.H. (2014) Transformation of the Linear Complementarity Problem and the Absolute Value Equation. *Journal of Jilin University (Science Edition)*, **4**, 638-686
- [13] Li, X.S. (1991) The Aggregation Function Method for Solving the Problem of Nonlinear Mini-Max Problem. *Computational Structural Mechanics and Its Application*, **8**, 85-92.
- [14] Chen, J.-S., Ko, C.-H. and Pan, S.H. (2010) A Neural Network Based on the Generalized Fischer-Burmeister Function for Nonlinear Complementarity Problems. *Information Sciences*, **180**, 697-711. <http://dx.doi.org/10.1016/j.ins.2009.11.014>
- [15] Ma, H.Q. and Huang, N.-J. (2014) Neural Network Smoothing Approximation Method for Stochastic Variational Inequality Problems. *Journal of Industrial and Management Optimization*, **7**, 645-660.
- [16] Liao, L.Z., Qi, H.D. and Qi, L.Q. (2001) Solving Nonlinear Complementarity Problems with Neural Networks: A Reformulation Method Approach. *Journal of Computational and Applied Mathematics*, **131**, 343-359. [http://dx.doi.org/10.1016/S0377-0427\(00\)00262-4](http://dx.doi.org/10.1016/S0377-0427(00)00262-4)
- [17] Miller, R.K. and Michel, A.N. (1982) Ordinary Differential Equations. Academic Press, Waltham.