Neural Networks 122 (2020) 40–53

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A smoothing neural network for minimization l_1-l_p in sparse signal reconstruction with measurement noises

You Zhao [a,b], Xing He [a,b,*], Tingwen Huang [c], Junjian Huang [b], Peng Li [d]

[a] Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China
[b] Key Laboratory of Machine Perception and Children's Intelligence Development, Chongqing University of Education, Chongqing, 400067, China
[c] Texas A & M University at Qatar, Doha 5825, Qatar
[d] Department of Electrical and Computer Engineering, Texas A & M University, College Station, TX77843, USA

ARTICLE INFO

Article history:
Received 19 November 2018
Received in revised form 27 August 2019
Accepted 8 October 2019
Available online 18 October 2019

Keywords:
Neural network
l_1-norm minimization
l_p-norm ($2 \geq p \geq 1$)
Smoothing approximation

ABSTRACT

This paper investigates a smoothing neural network (SNN) to solve a robust sparse signal reconstruction in compressed sensing (CS), where the objective function is nonsmooth l_1-norm and the feasible set satisfies an inequality of l_p-norm ($2 \geq p \geq 1$) which is used for measuring residual errors. With a smoothing approximate technique, the non-smooth and non-Lipschitz continuous issues of the l_1-norm and the gradient of l_p-norm can be solved efficiently. We propose a SNN which is modeled by a differential equation and give its circuit implementation. In this case, we prove the proposed SNN converges to the optimal of considered problem. Simulation results are discussed to demonstrate the efficiency of the proposed algorithm.

1. Introduction

Compressed sensing (CS) provides a new sampling paradigm for reducing data collection. On account of the pioneering works (Candes, 2008; Candes, Romberg, & Tao, 2006a, 2006b; Donoho, 2006), CS has become an active research branch in Bellasi and Benini (2015), Chen et al. (2011) and Kulkarni and Mohsenin (2017). CS is also an important tool and theory in signal processing. There are many tools and theories for signal processing, such as the linear canonical transform and the fractional Fourier transform (FRFT) (Wei, 2016, 2018; Wei & Li, 2016). The CS theory shows that the original sparse signal $x \in \mathbb{R}^n$ can be accurately reconstructed from a small number of linear measures $b = Ax \in \mathbb{R}^m$ ($m \ll n$), where $A \in \mathbb{R}^{m\times n}$ is the sensing matrix i.e., measurement matrix. Considering that the measurement noise is unavoidable in practical observation. Therefore, the compressed measure can be expressed as:

$$b = Ax + e$$

where $e \in \mathbb{R}^m$ is measurement noise.

Since $m \ll n$, reconstructing the original signal x from the compressed measurement b is generally ill-posed. Fortunately, Candes et al. (2006a) showed that, if sensing matrix A has some stable embedding properties, the original signal x can be reliably recovered with an error upper bounded which depends on the noise energy. Then, an natural method is to relax $Ax = b$ to yield the following optimization problem:

$$\min \ \|x\|_0 \ \ s.t. \ \|Ax - b\|_2 \leq \epsilon$$

where $\|\cdot\|_0$ is l_0-norm, $\epsilon > 0$ is upper bound of the residual error pre-determined from the noise level. However, solving the l_0 minimization problem is known to be NP-hard. Then, a convex relaxation model is the replacement for it to yield the following problem, i.e., BP denoising (BPDN) :

$$\min \ \|x\|_1 \ \ s.t. \ \|Ax - b\|_2 \leq \epsilon \tag{1}$$

According to the convex analysis theory, the problem (1) is equivalent to its Lagrangian version, i.e., the following l_1-l_2 minimization problem.

$$\min \ \|Ax - b\|_2^2 + \tau\|x\|_1 \tag{2}$$

where $\tau > 0$ be a regularization parameter which can be interpreted as a relative weight or trade-off between the residual error term $\|Ax-b\|_2^2$ and regularization term $\|x\|_1$. The l_1–l_2 minimization problem is more easily to solve because of its convexity, so it is widely used in many problems that can be converted in sparse signal reconstruction, such as data clustering (Elhamifar & Vidal, 2013), blind source separation (Li, Cichocki, & Amari, 2004, 2006), variable selection (Fan & Li, 2001) and face recognition (Wagner et al., 2012), etc. Some numerical iterative algorithms have been proposed for resolving l_1–l_2 minimization problem, such

* Corresponding author at: Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China.
E-mail addresses: Zhaoyou1991sdtz@163.com (Y. Zhao), hexingdoc@swu.edu.cn (X. He), tingwen.huang@qatar.tamu.edu (T. Huang).

https://doi.org/10.1016/j.neunet.2019.10.006
0893-6080/

as basis-pursuit (BP) (Candes et al., 2006b), Lasso (Tibshirani, 1996), the interior-point algorithm (Koh, Kim, & Boyd, 2007), augmented Lagrangian method (Afonso, Bioucas-Dias, & Figueiredo, 2011), Tomioka and Sugiyama (2009) and gradient projection method (Figueiredo, Nowak, & Wright, 2007), etc.

In paper Wen et al. (2017), a generalized l_p-norm ($0 \leqslant p < 2$) that can be viewed as the metric for the residual error was firstly proposed for investigating a robust sparse recovery with impulsive measurement noises in CS.

$$\min \|x\|_1 \quad s.t. \ \|Ax - b\|_p^p \leq \epsilon \tag{3}$$

where $\|x\|_p^p = \sum_{i=1}^n |x_i|^p$. Note that (1) is a special case of formulation (3), i.e., $p = 2$.

The intuition behind adopting l_p-norm ($2 \geq p > 1$) as the loss function is that, compared with the quadratic function, the l_p-norm has a less rapidly increasing function if $p < 2$. Therefore, the l_p-norm function has been used in many signal processing applications, such as array beamforming (Jiang et al., 2014), spectrum sensing (Moghimi, Nasri, & Schober, 2009), robust sparse signal recovery for l_p–l_1 minimization (Wen et al., 2017).

Since neural network methods can solve optimization problem efficiently by hardware circuit in parallel, they become a more and more hot topic in the branch of CS. Recently, a series of neural network algorithms and their applications have been reported in literature (Bian & Chen, 2012; Chen et al., 2011; Cheng, Hou, & Tan, 2009; Cheng et al., 2018; Dong & Zhu, 2018; Feng et al., 2017; Guo & Yang, 2015; Karimi & Gao, 2010; Leung, Sum, & Constantinides, 2014; Li & Wei, 2016a, 2016b; Liu & Hu, 2016; Liu & Wang, 2016; Wang et al., 2018; Wang & Zhang, 2017; Wei et al., 2018; Xu et al., 2012). Liu and Wang (Liu & Wang, 2016) investigated several one-layer projection neural network algorithms with lower model complexity for solving l_1-minimization problem based on a projection operator, but the proposed algorithms can only solve the problem of sparse signal reconstruction without noise measurement. A recurrent neural network (RNN) was proposed to handle constrained l_0-norm minimization by Guo and Yang in Guo and Yang (2015) based on an approximation method. It guarantees to obtain the globally convergent optimal solution. While its neurodynamic system can not handle the minimization problem with inequality constraints (with measurement noises). Based on the homotopy technique and the iterative hard thresholding (IHT) method, two homotopy methods for solving the l_0 minimization problem in CS are proposed by Dong and Zhu in Dong and Zhu (2018), which overcomes the difficulty of how to select parameters of the IHT method. However, it only effectively deals with the problem of sparse signal reconstruction without measurement noises. In addition, a one-layer smoothing neural network (SNN) algorithm to solve l_q ($1 > q > 0$) optimization problem was proposed by Bian and Chen (2012), which has a simple neural network model and just solve the linear constrained optimization problem (without measurement noises) effectively. Liu and Hu (2016) proposed an effective scaled gradient projection for solving l_1–l_2 minimization problem by use of a stationary penalty approach. Wang and Zhang (2017) adopted a smoothing projection neural network (SPNN) algorithm to solve a hybrid norm model l_{1-q} ($2 \geq q > 1$) minimization problem. The proposed SPNN algorithm can address the nonconvex hybrid norm optimization problem, while it cannot solve the nonlinear inequality constraint optimization problem (NICOP) (the problem with measurement noises) because the analytic expression of the projection operator of NICOP cannot be obtained. Based on the Fischer–Burmeister complementarity functions, Li and Wei presented a neural network method for the signal reconstruction by solving l_1–l_2 minimization problem (Li & Wei, 2016b). Later, two new Lagrange neural networks for sparse reconstruction problem by solving l_1–l_2 minimization problem (one is used for solving sparse signal reconstruction problem without measurement noise, the other is used to solve the same problem with measurement noise) and the circuit realization of proposed two neural networks are investigated in Li and Wei (2016a) by Li and Wei. Leung et al. (2014) and Feng et al. (2017) adopted Lagrange programming neural network (LMNN) for recovering sparse signal with noise-free and noisy measurements. The various Lagrange programming neural networks can effectively solve the l_1-norm minimization problem with Gaussian random noise, but they have some difficulties in solving l_q-norm ($1 > q > 0$) optimization problem or the sparse signal reconstruction problem with non-Gaussian noise measurement efficiently.

In paper Wen et al. (2017), the author investigated a L_p-ADM algorithm to solve the above problem (3). Considering that, neural network algorithms as an efficient method to recover sparse signal. However, there are few neural network algorithms for addressing the problem (3). In order to use a neural network algorithm to solve problem (3), a smoothing neural network based on a smoothing approximation method and multiplier method is researched in this paper. The main contributions of this paper are as follows:

(1) Instead of using differential inclusion approaches to deal with the non-differentiable optimization, a smooth approximation method is used to handle gradient of non-smooth function (l_1-norm) that can avoid the problem of parameter selection in differential inclusion (the derivative is zero) effectively in this paper.

(2) A smoothing approximation method is employed to make the gradients of l_q ($2 \geq q \geq 1$) to satisfy Lipschitz condition, which can solve the difficulties of non-Lipschitz gradient of l_q ($2 \geq q \geq 1$) in direct derivation method.

(3) Based on multiplier method, a differential equation is used to achieve the penalty parameter λ rather than fixed penalty parameter, which makes the proposed smoothing neural network (SNN) easily to implement by analog circuit.

(4) A Lyapunov method is applied to analyze and prove the stability of the proposed smoothing neural network algorithm.

The remainder of this paper is organized as follows. In Section 2, some definitions and lemmas of smoothing approximation are provided. In Section 3, the smoothing approximation model of $l_1 - l_p$ ($2 \geq p \geq 1$) minimization problem is introduced. In Section 4, a smoothing neural network is presented and its convergence and optimality are proved. In Section 5, three experimental results are described to illustrate effectiveness of the proposed SNN algorithm. An analog circuit implement of proposed SNN is presented in Section 6. Finally, we conclude the paper in Section 7.

Notations. Let column vectors $x = (x_1, x_2, \ldots, x_n)^T$ and $y = (y_1, y_2, \ldots, y_n)^T$, $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$ is the inner product of x and y, x_i is the ith element of x. $\|x\| = \left(\sum_{i=1}^n x_i^2\right)^{\frac{1}{2}}$ denotes the Euclidean norm. $\|x\|_p^p = \sum_{i=1}^n |x_i|^p$, ($2 \geq p \geq 1$), A_i is the i row of the matrix A, b_j is the jth element of vector b. $\nabla_x \hat{f}(x, \mu)$ means the gradient of $\hat{f}$ at x and $\nabla_\mu \hat{f}(x, \mu)$ represents the gradient of $\hat{f}$ at μ. The sign (x) is the signum function

$$\text{sign}(x) = \begin{cases} 1 & if \ x > 0 \\ [-1, \ 1] & if \ x = 0 \\ -1 & if \ x < 0 \end{cases}$$

2. Smoothing approximation

In the past decades, many smoothing approximation methods for solving nonsmooth optimizations have been developed in Bian and Chen (2014) and Chen (2012). The main feature of smoothing method is to approximate the nonsmooth function by the parametric smoothing function. The definition of smoothing function is as follows:

Definition 1. Let $\theta : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function. We call $\hat{\theta} : \mathbb{R}^n \times (0, +\infty) \to \mathbb{R}$ a smoothing function of θ, if $\hat{\theta}$ has the following conditions:

(1) For any fixed $\mu > 0$, $\hat{\theta}(\cdot, \mu)$ is continuous differentiable in $\mathbb{R}^n$, and for any fixed $x \in \mathbb{R}^n$, $\hat{\theta}(x, \cdot)$ is differentiable in $(0, +\infty)$.
(2) For any fixed $x \in \mathbb{R}^n$, $\lim_{\mu \to 0^+} \hat{\theta}(x, \mu) = \theta(x)$.
(3) There exists a positive constant $\kappa_{\hat{\theta}} > 0$ such that

$$\left|\nabla_\mu \hat{\theta}(x, \mu)\right| \leq \kappa_{\hat{\theta}} \quad \forall \mu \in (0, +\infty), \ x \in \mathbb{R}^n$$

(4) $\left\{\lim_{z \to x, \mu \to 0} \nabla_z \hat{\theta}(z, \mu)\right\} \subseteq \partial\theta(x)$

From the above definition (2) and (3), we deduce the following conditions:

$$\lim_{z \to x, \mu \to 0} \hat{\theta}(z, \mu) = \theta(x)$$
$$\left|\hat{\theta}(x, \mu) - \theta(x)\right| \leq \kappa_{\hat{\theta}} \mu \ \forall \mu \in [0, +\infty), \ x \in \mathbb{R}^n$$
$$\left|\hat{\theta}(x, \mu) - \hat{\theta}(x, \tilde{\mu})\right| \leq \kappa_{\hat{\theta}} (\mu - \tilde{\mu}), \forall x \in \mathbb{R}^n, \mu \geq \tilde{\mu} > 0$$

In the following, two propositions for the compositions of smoothing functions (Bian & Chen, 2012) are displayed:

Lemma 1. *(1) If $\hat{h}_1, \ldots, \hat{h}_m$ be smoothing functions of $h_1, \ldots, h_m$, then $\sum_{i=1}^m a_i \hat{h}_i$ is a smoothing function of $\sum_{i=1}^m a_i h_i$ with $k_{\sum_{i=1}^m a_i \hat{h}_i} = \sum_{i=1}^m a_i \kappa_{\hat{h}_i}$ when $a_i \geq 0$ and h_i is regular for any $i = 1, 2, \ldots, m$.*
(2) Let $\phi : \mathbb{R}^m \to \mathbb{R}$ be regular and $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable. Let $\hat{\varphi}$ be a smoothing function of ψ, then $\hat{\varphi}(\phi)$ is a smoothing function of $\varphi(\phi)$ with $\kappa_{\hat{\varphi}(\phi)} = \kappa_{\hat{\varphi}}$.
In this paper, the smooth approximation function with a simple structure (Chen, 2012) will be used.

$$\hat{\theta}(s, \mu) = \begin{cases} |s| & , \text{if } |s| > \mu \\ \frac{s^2}{2\mu} + \frac{\mu}{2} & , \text{if } |s| \leq \mu \end{cases} \tag{4}$$

where $\hat{\theta}(\cdot, \mu)$ is a convex of x for any $\mu > 0$, $\hat{\theta}(s, \cdot)$ is nondecreasing continuous function for any fixed $s \in \mathbb{R}$.

Next, two assumptions of the objective and constrained function are displayed.

Assumption 1. Both $f(x)$ and $g(x)$ functions are convex in $\mathbb{R}^n$.

Assumption 2 (*Slater Condition*)**.** There exists $\hat{x} \in \mathbb{R}^n$, such that $g(\hat{x}) < 0$.

3. The smoothing approximation to $l_1 - l_p$ ($2 \geq p \geq 1$) problem

$$\begin{aligned} &\min f(x) = \|x\|_1 \\ &s.t.\ g(x) = \|Ax - b\|_p^p - \epsilon \leq 0 \end{aligned} \tag{5}$$

where $\|\cdot\|_1$ is the 1-norm, $\|Ax - b\|_p^p = \sum_{i=j}^m \left|A_j x - b_j\right|^p$, $(2 \geq p \geq 1)$, and ϵ is a nonnegative parameter that can bound the p-norm of the residual error.

Next, we will show the problem (5) has the capability to successfully recover the original signals with a finite 2-norm error. A well-known condition of the sensing matrix A that be used to ensure the satisfactory recovery of x is called restricted isometry property (RIP) (Candes, 2008). Let integer $s = 1, 2, \ldots$, be the s-restricted isometry constant δ_s of A, the A satisfies the RIP condition if the following conditions are true

$$(1 - \delta_s)\|z\|^2 \leq \|Az\| \leq (1 + \delta_s)\|z\|^2$$

Lemma 2. *Supposing that the sensing matrix A satisfies the RIP of order $2s$ with $\delta_{2s} < \sqrt{2} - 1$. Then, for any measurement noise e with $\|e\|_p^p \leq \epsilon$ i.e., $\|e\|_p \leq \epsilon^{\frac{1}{p}}$, $1 \leq p \leq 2$, and any signal x supported on $\mathcal{T}_0$ with $|\mathcal{T}_0| \leq s$, the relation between the solutions x^* of problem (5) and true signal x^0 yields*

$$\|x^* - x^0\|_2 \leq C_s \epsilon^{\frac{1}{p}} \tag{6}$$

where $C_s = \frac{2\sqrt{2 + 2\delta_{2s}}}{1 - \left(1 + \sqrt{2}\right)\delta_{2s}}$.

Proof. The derivation is similar to that in Wen et al. (2017). Since x^* is a solution of problem (5) and the measurement noise obeys $\|e\|_p \leq \epsilon^{\frac{1}{p}}$ i.e., $\|Ax^* - b\|_p \leq \epsilon^{\frac{1}{p}}$, $\|Ax^0 - b\|_p \leq \epsilon^{\frac{1}{p}}$. Let $d = x^* - x^0$, we obtain

$$\begin{aligned} \|Ad\| &\leq \|Ax^* - b\|_2 + \|Ax^0 - b\|_2 \\ &\leq \|Ax^* - b\|_p + \|Ax^0 - b\|_p \\ &\leq 2\epsilon^{\frac{1}{p}} \end{aligned} \tag{7}$$

As the pointed out in Candes (2008) that

$$\|d\| = \frac{\sqrt{2 + 2\delta_{2s}}\|Ad\|_2}{1 - \left(1 + \sqrt{2}\right)\delta_{2s}} \tag{8}$$

if $\delta_{2s} < \sqrt{2} - 1$. Therefore, substituting (7) into (8), we obtain the conclusion. □

Next, the smoothing approximation of problem (5) based on smoothing functions presented in (4) will be discussed. Note that, $\|x\|_1$ is a convex, nonsmooth function, moreover, when $2 > p > 1$, the $g(x)$ is smooth and convex but its gradient is not Lipschitz continuous (not bounded), both the object functions $f(x)$ and constrains condition $g(x)$ are nonsmooth when $p = 1$. Therefore, the traditional proximal gradient methods cannot be used directly. To solve problem (5) using a neural network modeled as ordinary differential equation (ODE) (it is easy to implement by circuit), a smoothing approximation method (4) is adopted. Since $\lim_{z \to x, \mu \to 0} \hat{\theta}(z, \mu) = \theta(x)$, $f(x)$ can be approximated by

$$\hat{f}(x, \mu) = \sum_{i=1}^n \hat{f}(x_i, \mu), \ \hat{f}(x_i, \mu) = \begin{cases} |x_i| & , \text{if } |x_i| > \mu \\ \frac{x_i^2}{2\mu} + \frac{\mu}{2} & , \text{if } |x_i| \leq \mu \end{cases} \tag{9}$$

$g(x)$ is approximated by

$$\begin{aligned} &\hat{g}(x, \mu) = \sum_{j=m}^n w\left(A_j x - b_j\right) - \epsilon \leq 0 \\ &w\left(A_j x - b_j\right) = \begin{cases} \left|A_j x - b_j\right|^p & \text{if } \left|A_j x - b_j\right| > \mu \\ \left(\frac{\left(A_j x - b_j\right)^2}{2\mu} + \frac{\mu}{2}\right)^p & \text{if } \left|A_j x - b_j\right| \leq \mu \end{cases} \end{aligned} \tag{10}$$

Then, the problem (3) can be approached by the following smoothing function:

$$\min \hat{f}(x, \mu) \ s.t.\ \hat{g}(x, \mu) \tag{11}$$

4. Smoothing neural network

In this section, based on smoothing approximation method, we proposed a smoothing neural network algorithm to resolving problem (5).

$$\begin{cases} \dot{x} = -2\nabla_x \hat{f}(x, \mu) - 2\left[\lambda + \hat{g}(x, \mu)\right]^+ \nabla_x \hat{g}(x, \mu) \\ \dot{\lambda} = -\lambda + \left[\lambda + \hat{g}(x, \mu)\right]^+ \end{cases} \tag{12}$$

where λ is a penalty parameter, $[\cdot]^+ = \max\{0, \cdot\}$, $\mu = \mu_0 e^{-2t}$, $\nabla_x \hat{f}(x, \mu)$ is a gradient of the smoothing function of f with respect to x_i and it satisfies:

$$\nabla_{x_i} \hat{f}(x_i, \mu) = \begin{cases} sign(x_i) & , if\ |x_i| > \mu \\ \frac{x_i}{\mu} & , if\ |x_i| \le \mu \end{cases} \tag{13}$$

$\nabla_x \hat{g}(x, \mu)$ is the gradient of $\hat{g}(x, \mu)$ at x with any fix μ, which follows:

$$\nabla_x \hat{g}(x, \mu) = \sum_{j=1}^{m} \nabla_x w\left(A_j x - b_j\right)$$

$$\nabla_x w\left(A_j x - b_j\right) = \begin{cases} p\left|A_j x - b_j\right|^{p-1} sign\left(A_j x - b_j\right) A_j^T \\ \quad , if\ \left|A_j x - b_j\right| > \mu \\ p\left(\frac{\left(A_j x - b_j\right)^2}{2\mu} + \frac{\mu}{2}\right)^{p-1} \frac{\left(A_j x - b_j\right) A_j^T}{\mu} \\ \quad , if\ \left|A_j x - b_j\right| \le \mu \end{cases} \tag{14}$$

Theorem 1. *Suppose that Assumptions 1 and 2 hold, x^* is an optimal solution of problem (5) or (11) if and only if there exist $\lambda^* \in \mathbb{R}$, $\mu^* \to 0$ and the (x^*, λ^*, μ^*) satisfy:*

$$\begin{cases} 0 = -\nabla_x \hat{f}(x^*, \mu^*) - \left[\lambda^* + \hat{g}(x^*, \mu^*)\right]^+ \nabla_x \hat{g}(x^*, \mu^*) \\ 0 = -\lambda^* + \left[\lambda^* + \hat{g}(x^*, \mu^*)\right]^+ \end{cases} \tag{15}$$

Proof. According to the Karush–Kuhn–Tucker (KKT) conditions (Clarke, 1983), x^* is an optimal solution to problem (5), if and only if the following conditions hold:

$$0 \in \partial \|x^*\|_1 + \lambda^* \partial \|Ax^* - b\|_p^p \tag{16}$$

$$\|Ax^* - b\|_p^p - \epsilon \le 0 \tag{17}$$

$$\lambda^* \ge 0\ ,\ \lambda^*\left(\|Ax^* - b\|_p^p - \epsilon\right) = 0 \tag{18}$$

According to (4) in Definition 1, $\left\{\lim_{z \to x, \mu \to 0} \nabla_z \hat{\theta}(z, \mu)\right\} \subseteq \partial\theta(x)$, if $\mu \to 0^+$ and $\lambda^* = \left[\lambda^* + \hat{g}(x^*, \mu^*)\right]^+$.

We have

$$\begin{aligned} 0 &= -\nabla_x \hat{f}\left(x^*, \mu^*\right) - \left[\lambda^* + g\left(x^*\right)\right]^+ \nabla_x \hat{g}\left(x^*, \mu^*\right) \\ &\subseteq \partial \|x^*\|_1 + \lambda^* \partial \|Ax - b\|_p^p \end{aligned} \tag{19}$$

Then, the equality $\lambda^* = \left[\lambda^* + \hat{g}(x^*, \mu^*)\right]^+$ is true which shows that $\lambda^* \ge 0$, $\hat{g}(x^*, \mu^*) \le 0$ and $\lambda^* \hat{g}(x^*, \mu^*) = 0$. From the (2) of Definition 1 and $\mu^* \to 0^+$, one has $\lambda^* \ge 0$, $g(x^*) \le 0$ and $\lambda^* g(x^*) = 0$ which is equivalent to (17) and (18). The proof is complete. □

Theorem 2. *For any fixed $\mu_0 > \mu > 0$, There exists a unique solution of the smoothing neural network (12) if both $\nabla_x \hat{f}(\cdot, \mu)$ and $\nabla_x \hat{g}(\cdot, \mu)$ satisfy local Lipschitz condition.*

Proof. First, we show the $\nabla_x \hat{g}(x, \mu)$ at x for any fixed $\mu > 0$ satisfies Lipschitz condition, which can be derived by the global boundedness of the Clarke generalized gradient of $\nabla_x \hat{g}(x, \mu)$. According to the definition of Clarke gradient, $\nabla_x \hat{g}(x, \mu)$ is not differentiable when $\left|A_j x - b_j\right| = \mu$. Thus, we only need to prove the boundedness of the gradient of $\nabla_x \hat{g}(x, \mu)$ when $\left|A_j x - b_j\right| \ne \mu$.

For $\left|A_j x - b_j\right| > \mu$, with fixed $\mu > 0$, we obtain

$$\begin{aligned} \left|\nabla_x^2 w\left(A_j x - b_j, \mu\right)\right| &= \left|p(p-1)\left|A_j x - b_j\right|^{p-2} A_j A_i^T\right| \\ &\le p\left|A_j A_i^T\right| \mu^{p-2} \end{aligned} \tag{20}$$

when $\left|A_j x - b_j\right| < \mu$, one has

$$\begin{aligned} &\left|\nabla_x^2 w\left(A_j x - b_j, \mu\right)\right| \\ =& p(p-1)\left(\frac{\left(A_j x - b_j\right)^2}{2\mu} + \frac{\mu}{2}\right)^{p-2} \frac{\left(A_j x - b_j\right)^2 A_j A_i^T}{\mu} \\ &+ p\left(\frac{\left(A_j x - b_j\right)^2}{2\mu} + \frac{\mu}{2}\right)^{p-1} \frac{A_j A_i^T}{\mu} \\ \le& p(p-1)\left(\frac{\mu}{2}\right)^{p-2} \frac{\left|A_j A_i^T\right|}{\mu} + p\mu^{p-1} \frac{\left|A_j A_i^T\right|}{\mu} \\ \le& 2p \frac{\left|A_j A_i^T\right|}{\mu}\left(\mu^{p-2} + \mu^{p-1}\right) \end{aligned} \tag{21}$$

Therefore, for any $x, r \in \mathbb{R}^n$, we have

$$\begin{aligned} &\|\nabla_x \hat{g}(x, \mu) - \nabla_r \hat{g}(r, \mu)\| \\ \le& 2np \frac{\max_{i,j} \left|A_j A_i^T\right|}{\mu}\left(\mu^{p-2} + \mu^{p-1}\right)\|x - r\| \end{aligned} \tag{22}$$

The same way for $\nabla_x \hat{g}(x, \mu)$, for any $x, r \in \mathbb{R}^n$, one has

$$\|\nabla_x \hat{f}(x, \mu) - \nabla_r \hat{f}(r, \mu)\| \le \frac{n}{\mu}\|x - r\| \tag{23}$$

Denote $z_1 = \left[x_1^T, \lambda_1\right]^T \in \mathbb{R}^{n+1}$ and $z_2 = \left[x_2^T, \lambda_2\right]^T \in \mathbb{R}^{n+1}$ are two solutions of SNN with initial point $z_0 \in \mathbb{R}^{n+1}$. Suppose there exists a $\hat{t}$, such that $z_1(\hat{t}) \ne z_2(\hat{t})$. There exists a $\varrho > 0$, such that $z_1(t) \ne z_2(t)$, $\forall t \in \left[\hat{t}, \hat{t} + \varrho\right]$. Write

$$\Phi(z, \mu) = \begin{pmatrix} -2\nabla_x \hat{f}(x, \mu) - 2\left[\lambda + \hat{g}(x, \mu)\right]^+ \nabla_x \hat{g}(x, \mu) \\ -\lambda + \left[\lambda + \hat{g}(x, \mu)\right]^+ \end{pmatrix} \tag{24}$$

For any fixed $\mu > 0$, $\nabla_x \hat{f}(x, \mu)$ and $\nabla_x \hat{g}(x, \mu)$ satisfies Lipschitz condition, so is $\Phi(z, \mu)$. Therefore, there exists a $L > 0$, such that

$$\|\Phi(z_1, \mu) - \Phi(z_2, \mu)\| \le L\|z_1 - z_2\| \tag{25}$$

which yields

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}\|z_1 - z_2\|^2 &= (z_1 - z_2)^T(\dot{z}_1 - \dot{z}_2) \\ &= (z_1 - z_2)^T(\Phi(z_1, \mu) - \Phi(z_2, \mu)) \\ &\le L\|z_1 - z_2\|^2\ , \forall t \in \left[0, \hat{t} + \varrho\right] \end{aligned} \tag{26}$$

Integrating the above inequality form 0 to t $\left(t \le \hat{t} + \varrho\right)$ and applying Gronwall's inequality, we obtain that $z_1 = z_2$. The proof is complete. □

Theorem 3. *With Assumptions 1 and 2, the trajectory $\left[x^T, \lambda\right]^T \in \mathbb{R}^{n+1}$ associated with SNN (12) is stable and x converges to an optimal solution of problem (5) with any given initial point.*

Proof. Let $x^* \in \mathbb{R}^n$ be an optimal solution of problem (5). From Definition 1, there exists $\lambda^* \in \mathbb{R}$ and μ^* satisfies the equations in (10)–(12).

Considering the following Lyapunov function:

$$\begin{aligned}
&V(x,\lambda)\\
=&\hat{f}(x,\mu)-\hat{f}\left(x^*,\mu^*\right)-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)\left(\mu-\mu^*\right)\\
&+\frac{1}{2}\left(\|\left[\lambda+\hat{g}(x,\mu)\right]^+\|^2-\|\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\|^2\right)+k_{\hat{f}}\mu\\
&-\left(x-x^*\right)^T\left(\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right.\\
&\left.+\nabla_x\hat{f}\left(x^*,\mu^*\right)\right)+\frac{1}{2}\|x-x^*\|^2+\frac{1}{2}\|\lambda-\lambda^*\|^2\\
&-\left(\lambda-\lambda^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+ +k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\
&-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left(\mu-\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+
\end{aligned}\tag{27}$$

where $k_{\hat{f}}=k_{\sum_{i=1}^n\hat{f}\left(x_i,\mu\right)}>0$, $k_{\hat{g}}=k_{\sum_{i=1}^m w\left(A_jx-b_j\right)}>0$ and $\mu=\mu_0e^{-2t}$.

Taking the derivative of $V(x,\lambda)$ with respect to t, we have

$$\begin{aligned}
&\frac{dV(x,\lambda)}{dt}\\
\le&\sup_{\substack{\nabla_x\hat{f}(x,\mu)\\ \nabla_x\hat{g}(x,\mu)}}2\left(\nabla_x\hat{f}(x,\mu)+\nabla_x\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+\right.\\
&\left.+x-x^*-\nabla_x\hat{f}\left(x^*,\mu\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\\
&\times\left(-\nabla_x\hat{f}(x,\mu)-\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)\right)-2k_{\hat{f}}\mu\\
&+\left(\left[\lambda+\hat{g}(x,\mu)\right]^+-\left[\lambda^*+\hat{g}\left(x^*,\mu\right)\right]^++\lambda-\lambda^*\right)^T\\
&\times\left(-\lambda+\left[\lambda+\hat{g}(x,\mu)\right]^+\right)-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\
&-2\mu\left(\nabla_\mu\hat{f}(x,\mu)+\nabla_\mu\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+\right.\\
&\left.-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)=\mathcal{V}
\end{aligned}\tag{28}$$

In what follows, we will show some properties of $\mathcal{V}$ i.e., the right-hand side of (28).

$$\begin{aligned}
\mathcal{V}=&2\left(\nabla_x\hat{f}(x,\mu)+\nabla_x\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+ +x-x^*\right.\\
&\left.-\nabla_x\hat{f}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\\
&\times\left(x-x^*-\nabla_x\hat{f}(x,\mu)-\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)\right)\\
&+2\left(\nabla_x\hat{f}(x,\mu)+\nabla_x\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+ +x-x^*\right.\\
&\left.-\nabla_x\hat{f}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\\
&\times\left(x^*-x\right)-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\
&+\left(\left[\lambda+\hat{g}(x,\mu)\right]^+-\left[\lambda^*+\hat{g}\left(x^*,\mu\right)\right]^++\lambda-\lambda^*\right)^T\\
&\times\left(-\lambda+\left[\lambda+\hat{g}(x,\mu)\right]^+\right)-2k_{\hat{f}}\mu\\
&-2\mu\left(\nabla_\mu\hat{f}(x,\mu)+\nabla_\mu\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+\right.\\
&\left.-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)
\end{aligned}\tag{29}$$

let

$$\begin{aligned}
F_1=&2\left(\nabla_x\hat{f}(x,\mu)+\nabla_x\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+ +x-x^*\right.\\
&\left.-\nabla_x\hat{f}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+g\left(x^*\right)\right]^+\right)^T\\
&\times\left(-\nabla_x\hat{f}(x,\mu)-\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)+x-x^*\right)
\end{aligned}\tag{30}$$

and

$$\begin{aligned}
F_2=&2\left(\nabla_x\hat{f}(x,\mu)-\nabla_x\hat{f}\left(x^*,\mu^*\right)+x-x^*\right)^T\\
&\times\left(x^*-x\right)-2k_{\hat{f}}\mu-2\mu\left(\nabla_\mu\hat{f}(x,\mu)-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)\right)
\end{aligned}\tag{31}$$

and

$$\begin{aligned}
F_3=&\left(\left[\lambda+\hat{g}(x,\mu)\right]^+-\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^++\lambda-\lambda^*\right)^T\\
&\times\left(-\lambda+\left[\lambda+\hat{g}(x,\mu)\right]^+\right)-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\
&+2\left(\nabla_x g(x)\left[\lambda+g(x)\right]^+-\nabla_x g\left(x^*\right)^T\left[\lambda^*+g\left(x^*\right)\right]^+\right)^T\\
&\times\left(x^*-x\right)-2\mu\left(\nabla_\mu\hat{g}(x,\mu)\left[\lambda+\hat{g}(x,\mu)\right]^+\right.\\
&\left.-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)
\end{aligned}\tag{32}$$

For F_1, we get

$$\begin{aligned}
F_1=&-2\left(\nabla_x\hat{f}\left(x^*,\mu^*\right)+\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\\
&\times\left(-\nabla_x\hat{f}(x,\mu)-\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)+x-x^*\right)\\
&+2\left(\nabla_x\hat{f}(x,\mu)+\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)+x-x^*\right)^T\\
&\times\left(-\nabla_x\hat{f}(x,\mu)-\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)+x-x^*\right)\\
=&2\|x-x^*\|^2-2\|\nabla_x\hat{f}(x,\mu)+\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)\|^2
\end{aligned}\tag{33}$$

Since x^* is an optimal solution of problem (5), then $\nabla_x\hat{f}\left(x^*,\mu^*\right)+\left[\lambda^*+g\left(x^*\right)\right]^+\nabla_x g\left(x^*\right)=0$. Therefore, we have

$$F_1=2\|x-x^*\|^2-2\|\nabla_x\hat{f}(x,\mu)+\left[\lambda+\hat{g}(x,\mu)\right]^+\nabla_x\hat{g}(x,\mu)\|^2\tag{34}$$

For F_2, one have

$$\begin{aligned}
F_2=&2\left(\nabla_x\hat{f}(x,\mu)-\nabla_x\hat{f}\left(x^*,\mu^*\right)+x-x^*\right)^T\\
&\times\left(x^*-x\right)-2\mu\left(\nabla_\mu\hat{f}(x,\mu)-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)\right)-2k_{\hat{f}}\mu\\
=&-2\|x-x^*\|^2+2\left(\nabla_x\hat{f}(x,\mu)-\nabla_x\hat{f}\left(x^*,\mu^*\right)\right)^T\\
&\times\left(x^*-x\right)-2k_{\hat{f}}\mu-2\mu\left(\nabla_\mu\hat{f}(x,\mu)-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)\right)\\
\le&-2\|x-x^*\|^2
\end{aligned}\tag{35}$$

Utilizing the convexity of $\hat{f}(x,\mu)$ about x and μ provided in Appendix, we have

$$\begin{aligned}
&-k_{\hat{f}}\left(\mu-\mu^*\right)+\nabla_x\hat{f}\left(x^*,\mu^*\right)\left(x-x^*\right)\leqslant\hat{f}(x,\mu)-\hat{f}\left(x^*,\mu^*\right)\\
&\leqslant k_{\hat{f}}\left(\mu-\mu^*\right)+\nabla_x\hat{f}(x,\mu)\left(x-x^*\right)
\end{aligned}\tag{36}$$

then

$$\left(\nabla_x \hat{f}\left(x,\mu\right)-\nabla_x \hat{f}\left(x^*,\mu^*\right)\right)^T\left(x^*-x\right)+2k_{\hat{f}}\left(\mu^*-\mu\right)\leqslant 0 \tag{37}$$

Since $\mu\geq\mu^*>0$, $\nabla_\mu \hat{f}\left(x,\mu\right)\geqslant 0$, $\nabla_\mu \hat{f}\left(x^*,\mu^*\right)=0$, we have

$$\begin{aligned}&\left(\nabla_x \hat{f}\left(x,\mu\right)-\nabla_x \hat{f}\left(x^*,\mu^*\right)\right)^T\left(x^*-x\right)-2k_{\hat{f}}\mu\\&-2\mu\left(\nabla_\mu \hat{f}\left(x,\mu\right)-\nabla_\mu \hat{f}\left(x^*,\mu^*\right)\right)\leqslant 0\end{aligned} \tag{38}$$

For F_3, we have

$$\begin{aligned}F_3=&\left(\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+-2\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+ +\lambda\right)^T\\&\times\left(-\lambda+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right)\\&+2\left(\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\nabla_x\hat{g}\left(x,\mu\right)\right.\\&\left.-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\left(x^*-x\right)\\&-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\&-2\mu\left(\nabla_\mu\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right.\\&\left.-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)\\=&-\left\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]-\lambda\right\|^2-2\mu k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\\&+2\left(\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+-\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\\&\times\left(-\lambda+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right)\\&+2\left(\nabla_x\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right.\\&\left.-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)^T\left(x^*-x\right)\\&-2\mu\left(\nabla_\mu\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right.\\&\left.-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)\\=&-\left\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]-\lambda\right\|^2+2\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\\&\times\left(-\lambda+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+ +\nabla_x\hat{g}\left(x,\mu\right)\left(x^*-x\right)\right)\\&-2\left(-\lambda+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+ +\nabla_x\hat{g}\left(x^*,\mu^*\right)^T\left(x^*-x\right)\right)\\&\times\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+-2uk_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\\&-2\mu\left(\nabla_\mu\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right.\\&\left.-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)\end{aligned} \tag{39}$$

Using the following conditions:

$$\begin{aligned}&\lambda+\hat{g}\left(x,\mu\right)=\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+-\left[-\lambda-\hat{g}\left(x,\mu\right)\right]^+\\&-\lambda+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+=\hat{g}\left(x,\mu\right)+\left[-\lambda-\hat{g}\left(x,\mu\right)\right]^+\end{aligned} \tag{40}$$

we have

$$\begin{aligned}F_3=&-\left\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+-\lambda\right\|^2+2\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\\&\times\left(\hat{g}\left(x,\mu\right)-\hat{g}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x,\mu\right)\left(x-x^*\right)\right)\\&+2\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\left(\hat{g}\left(x^*,\mu^*\right)+\left[-\lambda-\hat{g}\left(x,\mu\right)\right]^+\right)\\&-2uk_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+-2\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\\&\times\left(\hat{g}\left(x,\mu\right)-\hat{g}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)^T\left(x-x^*\right)\right)\\&-2\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\left(\hat{g}\left(x^*,\mu^*\right)+\left[-\lambda-\hat{g}\left(x,\mu\right)\right]^+\right)\\&-2\mu\left(\nabla_\mu\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\right.\\&\left.-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)\end{aligned} \tag{41}$$

According to the convexity of $\hat{g}\left(x,\mu\right)$ of x and μ that can be seen in Appendix, we have

$$\begin{aligned}&\hat{g}\left(x,\mu\right)-\hat{g}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x,\mu\right)\left(x-x^*\right)\\\leq&\nabla_\mu\hat{g}\left(x,\mu\right)\left(\mu-\mu^*\right)\\&\left(\hat{g}\left(x,\mu\right)-\hat{g}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left(x-x^*\right)\right)\\\geq&\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left(\mu-\mu^*\right)\end{aligned} \tag{42}$$

Since $\nabla_\mu\hat{g}\left(x,\mu\right)\geq 0$, we get

$$\begin{aligned}&2\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\left(\hat{g}\left(x,\mu\right)-\hat{g}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x,\mu\right)\left(x-x^*\right)\right)\\&-2\nabla_\mu\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\mu\\\leq&2\nabla_\mu\hat{g}\left(x,\mu\right)\left(\mu-\mu^*\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\\&-2\nabla_\mu\hat{g}\left(x,\mu\right)\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\mu\\=&-2\nabla_\mu\hat{g}\left(x,\mu\right)\mu^*\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\leq 0\end{aligned} \tag{43}$$

and

$$\begin{aligned}&-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+u-2\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\\&\times\left(\hat{g}\left(x,\mu\right)-\hat{g}\left(x^*,\mu^*\right)-\nabla_x\hat{g}\left(x^*,\mu^*\right)\left(x-x^*\right)\right)\\&+2\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\\leq&-2\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left(\mu-\mu^*\right)\\&+2\mu\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\\&-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\\\leq&2\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\mu^*\\&-2k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\leq 0\end{aligned}$$

Since x^* is an optimal solution of problem (11), one has $\hat{g}\left(x^*,\mu^*\right)\leq 0$, then $\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\hat{g}\left(x^*,\mu^*\right)\leq 0$, $\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\hat{g}\left(x^*,\mu^*\right)\leq 0$, and $\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\left[-\lambda-\hat{g}\left(x,\mu\right)\right]^+=0$. Then, we have

$$F_3\leq-\left\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+-\lambda\right\|^2 \tag{44}$$

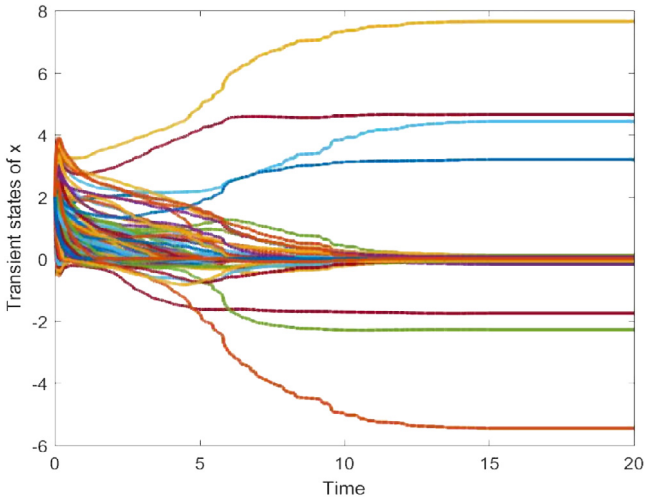

Fig. 1. The transient behaviors of SNN with p = 1.5 under SNR = 30 dB.

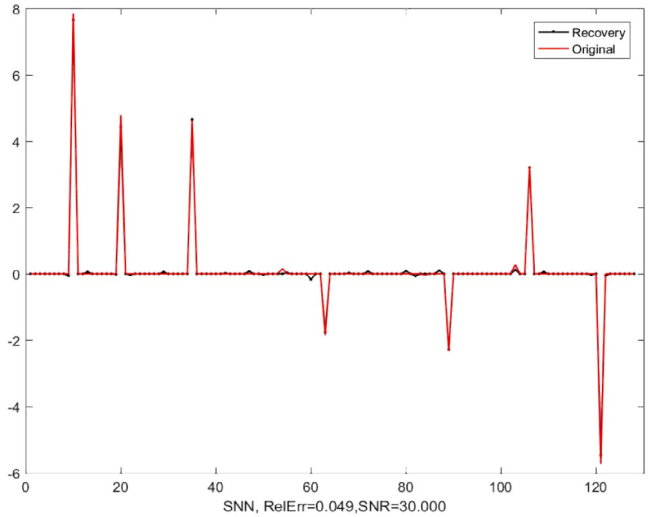

Fig. 2. The recovered sparse signal of SNN with p = 1.5, RelErr=0.049 under SNR = 30 dB.

Therefore

$$\begin{aligned} &F_1+F_2+F_3\\ \le&-\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+-\lambda\|^2\\ &-2\|\nabla_x\hat{f}\left(x,\mu\right)+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\nabla_x\hat{g}\left(x,\mu\right)\|^2 \end{aligned} \tag{45}$$

Combining with (25) implies

$$\begin{aligned} &\frac{dV\left(x,\lambda\right)}{dt}\\ \le&-\inf_{\substack{\nabla_x\hat{f}\left(x,\mu\right)\\ \nabla_x\hat{g}\left(x,\mu\right)}}\Big\{\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]-\lambda\|^2\\ &+2\|\nabla_x\hat{f}\left(x,\mu\right)+\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\nabla_x\hat{g}\left(x,\mu\right)\|^2\Big\} \end{aligned} \tag{46}$$

Obviously, the convexity of $\hat{g}\left(x,\mu\right)$ and $\hat{f}\left(x,\mu\right)$ of x and μ, it follows that

$$\begin{aligned} &\hat{f}\left(x,\mu\right)-\hat{f}\left(x^*,\mu^*\right)-\nabla_\mu\hat{f}\left(x^*,\mu^*\right)\left(\mu-\mu^*\right)\\ &+\frac{1}{2}\left(\|\left[\lambda+\hat{g}\left(x,\mu\right)\right]^+\|^2-\|\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\|^2\right)\\ &-\left(\lambda-\lambda^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+-\left(x-x^*\right)^T\\ &\times\left(\nabla_x\hat{f}\left(x^*,\mu^*\right)+\nabla_x\hat{g}\left(x^*,\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\right)\\ &-\nabla_\mu\hat{g}\left(x^*,\mu^*\right)\left(\mu-\mu^*\right)\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\ge 0 \end{aligned} \tag{47}$$

and $k_{\hat{f}}\mu+k_{\hat{g}}\left[\lambda^*+\hat{g}\left(x^*,\mu^*\right)\right]^+\mu\ge 0$, we have

$$\begin{aligned} \frac{L+1}{2}\|x-x^*\|^2+\frac{L+1}{2}\|\lambda-\lambda^*\|^2&\ge V\left(x,\lambda\right)\\ &\ge\frac{1}{2}\|x-x^*\|^2+\frac{1}{2}\|\lambda-\lambda^*\|^2 \end{aligned} \tag{48}$$

Therefore, the SNN converges to its equilibrium points i.e., the optimal solution of our considered problem. The proof is complete. □

Remark 1. If $p=2$, the proposed SNN can be replaced by Eq. (49) and its stability can be proved analogous to the above methods:

$$\begin{cases} \dot{x}=-2\nabla_x\hat{f}\left(x,\mu\right)-2\left[\lambda+g\left(x\right)\right]^+\nabla_x g\left(x\right)\\ \dot{\lambda}=-\lambda+\left[\lambda+g\left(x\right)\right]^+ \end{cases} \tag{49}$$

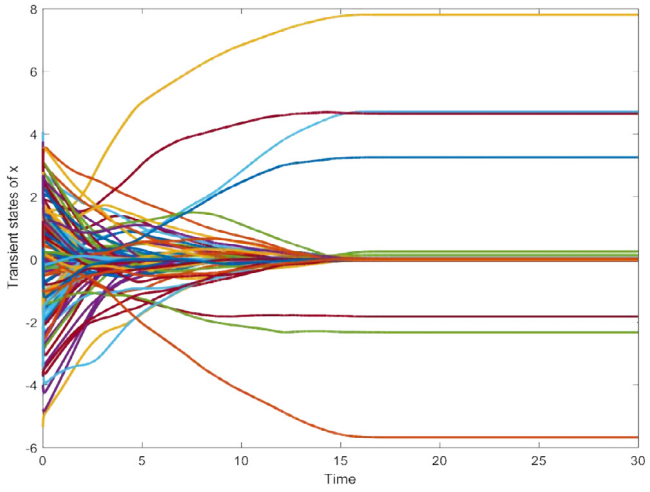

Fig. 3. The transient behaviors of SNN with p = 1.5 under SNR = 40 dB.

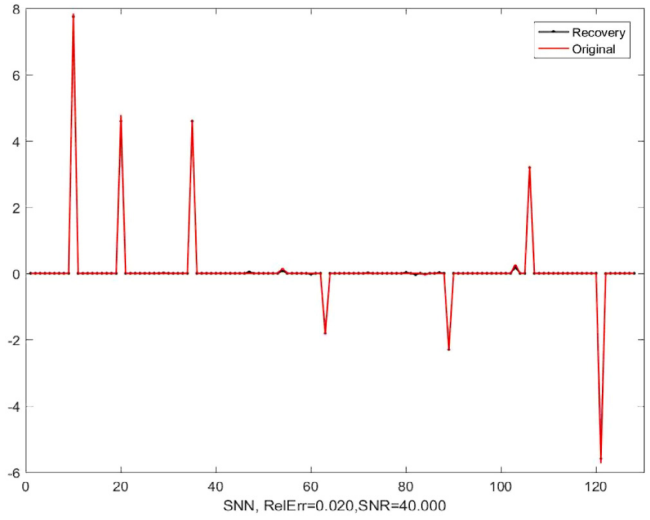

Fig. 4. The recovered sparse signal of SNN with p = 1.5, RelErr = 0.049 under SNR = 40 dB.

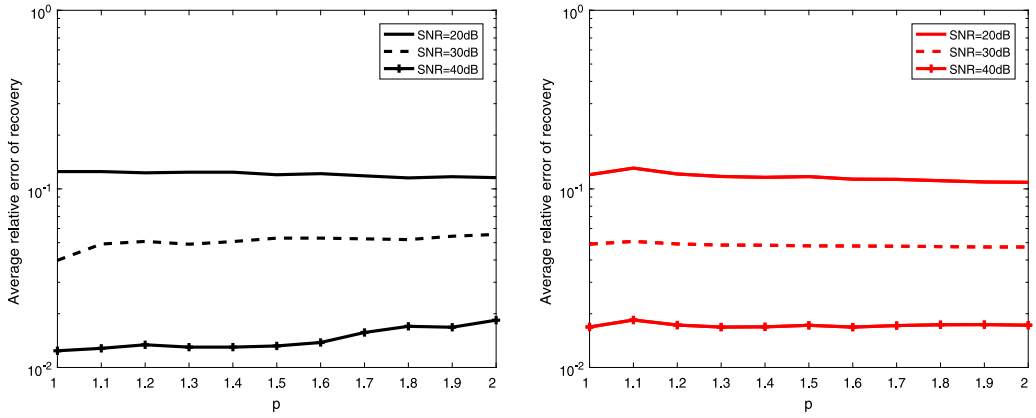

Fig. 5. Recovery performance of SNN with various p = 1.1, 1.2, …, 2 in Gaussian noise (left) and Recovery performance of L_p-ADM (Wen et al., 2017) with various p = 1.1, 1.2, …, 2 in Gaussian noise..

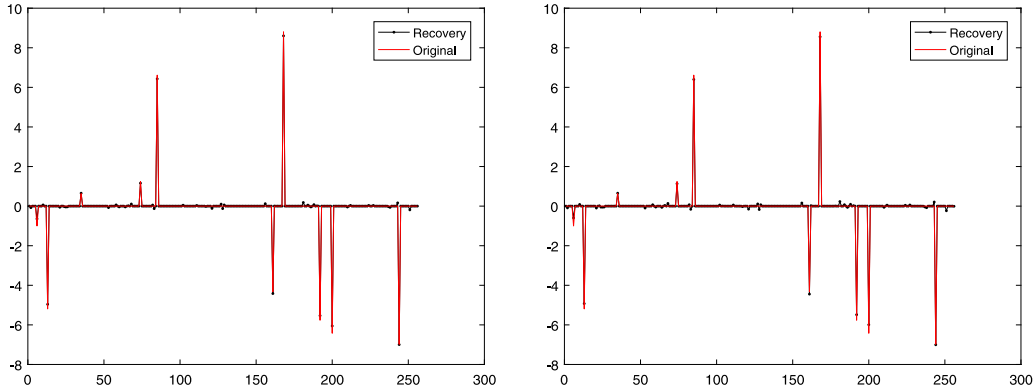

Fig. 6. The recovered sparse signal of SNN with Err =0.8375 (left) and the recovered sparse signal of LPNN (right) with Err = 1.0151, under $m = 64$, $n = 256$, $S = 10$ and $\sigma^2 = 0.05$.

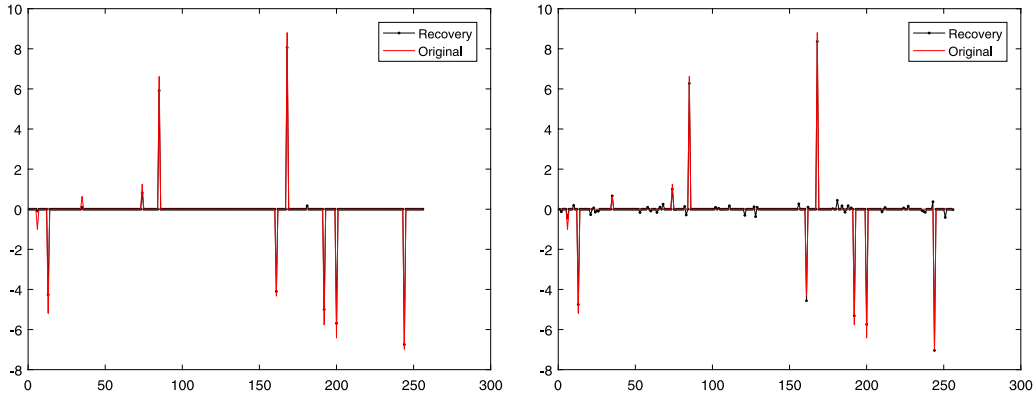

Fig. 7. The recovered sparse signal of BPDN with Err = 1.113 (left) and the recovered sparse signal of SGP(right) with Err = 1.4557, under $m = 64$, $n = 256$, $S = 10$ and $\sigma^2 = 0.05$.

5. Numerical simulations

In this section, three experiments will be to done to illustrate the performance of SNN (12). The following three steps are needed to get the sensing matrix A and measurement vector b.

(1) Generating a random S sparse signal vector $x \in \mathbb{R}^n$, i.e., there exists a random S nonzero components in vector x.

(2) Getting the entries of sensing matrix A with normal Gaussian distribution.

(3) Computing the measurement vector b by using equation "Ax"+"noises".

Example 1. In this experiment, A random Gaussian matrix, where $m = 40$, $n = 128$ the sparsity $S = 10$ and the locations of which are randomly determined, is used as the sensing matrix. The noise is appropriately added to generate noisy measurements according to desired noise levels. The desired signal-to-noise ratio (SNR), measured in decibel (dB) is as follows (Wen et al., 2017):

$$\mathrm{SNR} = 20\log_{10}\left(\frac{\|Ax^0 - E\{Ax^0\}\|}{\|e\|}\right) \tag{50}$$

First, we evaluate the convergent behavior of SNN when $p = 1.5$ with Gaussian noise at various noise levels. The performance is evaluated by relative error as $\mathrm{RelErr} = \|x^0 - x^*\|/\|x^0\|$, where

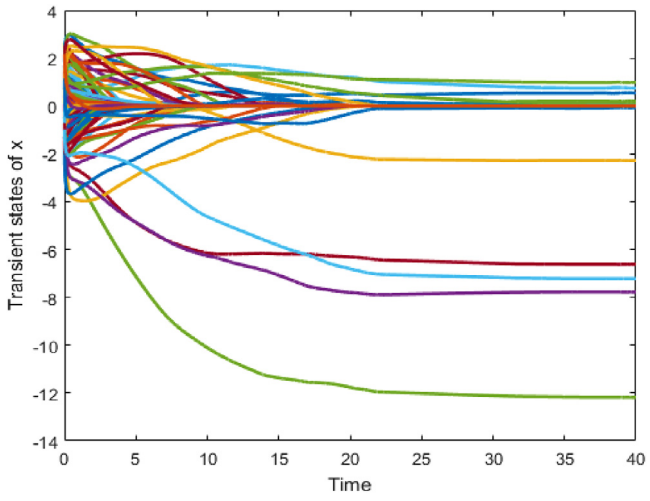

Fig. 8. The transient behaviors of SNN with $p = 1$.

x^* is the recovered value of the true signal x^0. As can be see from Figs. 1, 3, the state curves of SNN converge to the KKT stable points as t increases. The recovery performance signals and the RelErr are displayed in Figs. 2, 4. As shown in Figs. 2, 4, the recovered signal is close to the original signal with a certain RelErr at different SNR levels, which coincides with the conclusion presented in Lemma 2. Next, we investigate the recovery performance of SNN versus p in Gaussian noise with 20 dB, 30 dB and 40 dB by calculating the averaged relative error (50times). In Fig. 5, we investigate the averaged relative error (50times) of SNN and $L_p - ADM$ (Wen et al., 2017), and its results show that the SNN algorithm is quite similar to the $L_p - ADM$.

Example 2. In this experiment, we also use the orthonormal Gaussian random matrix with $m = 64$, $n = 256$ as a sensing matrix. With the sparsity $S = 10$ and the upper bound of the residual error $\epsilon = m\sigma^2$, we evaluate the effectiveness of SNN algorithm (49) (by using Err $= \|x - x^*\|$, x^* is the recovered values of the true signal x^0). Compared with the state-of-the-art algorithms: Basis Pursuit De-Noising (BPDN), LPNN (Feng et al., 2017), the scaled gradient projection (SGP) (Liu & Hu, 2016) with $\sigma^2 = 0.05$. It can be see from Fig. 6, the SNN has a smaller Err than LPNN with $\sigma^2 = 0.005$. From Fig. 7, the BPDN has less Err than SGP. Thus, the SNN algorithm a better performance than other algorithms, the BPDN and LPNN algorithms have a similar performance and the SGP algorithm has the worst performance than SNN, LPNN, BPDN.

Example 3. In this example, we consider a practical condition that the measurements are contaminated by bit error like corruption, which gives rise to larger errors in measurements. We take the length 128 with 10 nonzero values as original sparse signal and use orthonormal Gaussian random matrix with 40 observations as the sensing matrix. Ten percent of the measurements are set ± 500 randomly and zero-mean Gaussian noise with variance 0.01 is to imitate the background noise, the valuation of ϵ is $500 \times 0.1 \times 40$. Figs. 8 and 9 show the transient behaviors of SNN ($p = 1$) and the recovery performance of the compared robust algorithms with the relative error of recovery (RelErr). From Fig. 8, we can see that the transient behaviors of SNN ($p = 1$) are asymptotic stability. From Fig. 9, we can see that the SNN ($p = 1$) has relatively high robustness than Lasso, L_q-min Foucart and Lai (2009), BP-SEP (Studer & Baraniuk, 2014) algorithms and has relatively low robustness than L_p-ADM. This is because that the L_p-ADM algorithm adopts different proximity operators for different p values (for example using a soft-threshold method when $p = 1$), so it has better robustness than SNN algorithm, which only use the same approximation method for different p values.

6. Circuit implementation of SNN (12)

In this section, an analog circuit implementation of SNN (12) is proposed. First, some elementary circuits needed in the complete circuit are presented in Figs. 10–20. Fig. 10 shows a circuit to carry out the operation of $V_{out} = \max\{0, V_{in}\}$ by using two Op Amps and a diode. Fig. 11 displays a circuit which get $V_{out} = |V_{in}|$ operation with two Op Amps, two diodes and five resistors. Fig. 12 shows two voltage selection circuits by using two

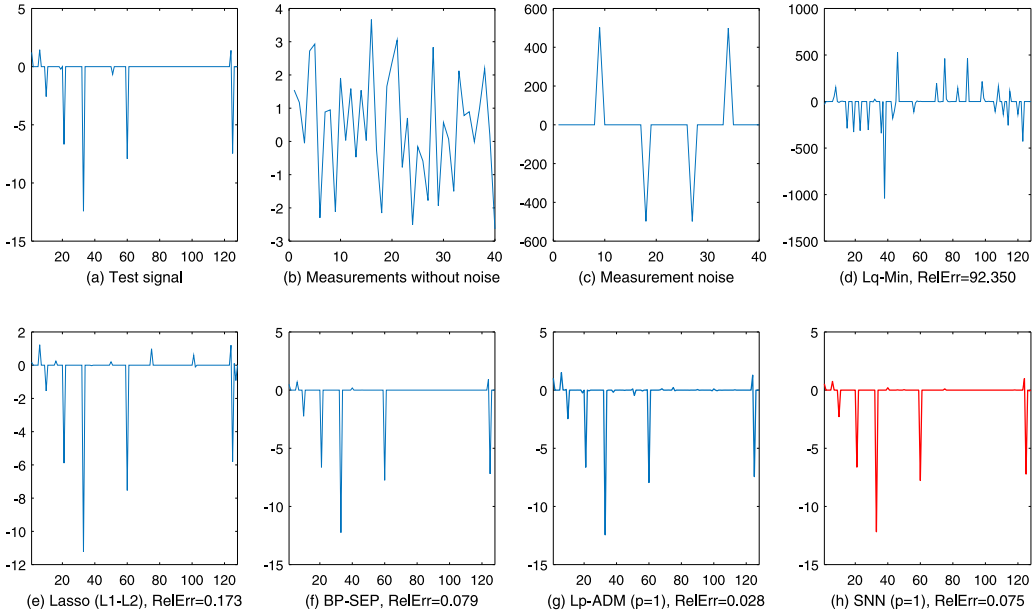

Fig. 9. The recovery performance of compared algorithms in the presence of bit error. (a) Test signal. (b) Measurements without noise. (c) Measurement noise. (d) Lq-Min, RelErr = 92.350. (e) Lasso (L1–L2), RelErr = 0.173. (f) BP-SEP, RelErr = 0.0797. (g) Lp-ADM (p = 1), RelErr = 0.028. (h) SNN (p = 1), RelErr = 0.075.

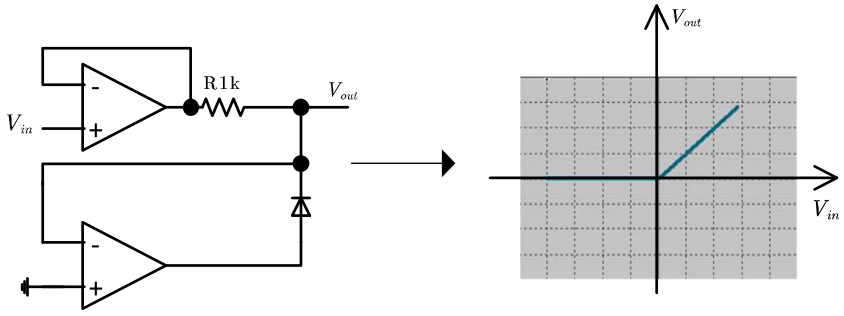

Fig. 10. Circuit implementation of term $[\cdot]^+ = \max\{0, \cdot\}$.

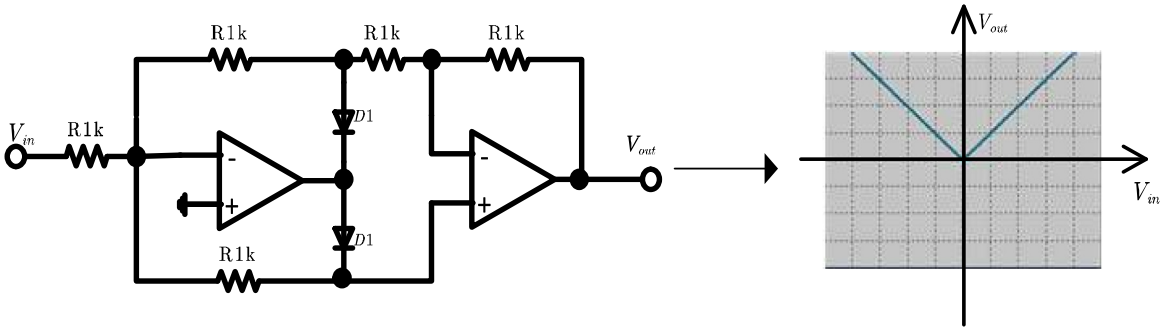

Fig. 11. Circuit implementation of term $|\cdot|$.

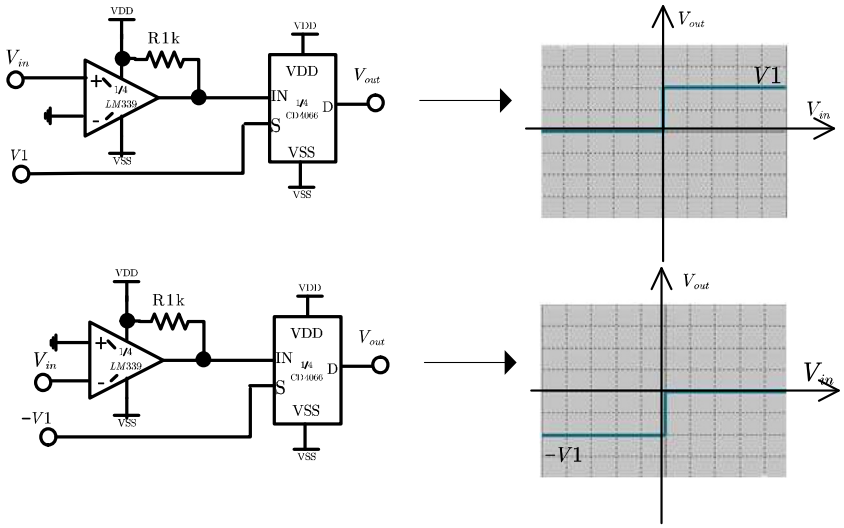

Fig. 12. Circuit implementation of selective circuit. Top: $V_{out} = V_1$, if $V_{in} \geq 0$ and $V_{out} = 0$, otherwise. Bottom: $V_{out} = 0$, if $V_{in} \geq 0$ and $V_{out} = -V_1$, otherwise.

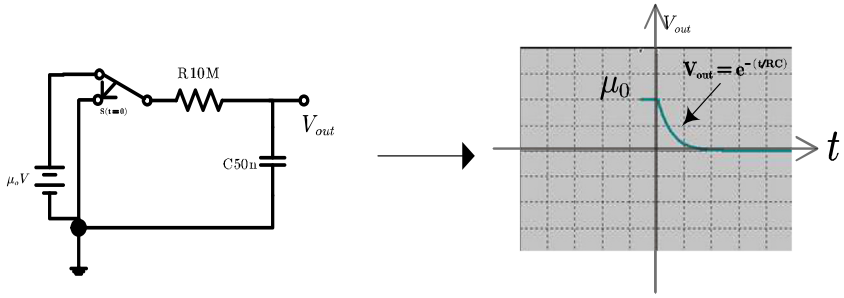

Fig. 13. Circuit implementation of term $u = u_0 e^{-t/(RC)}$.

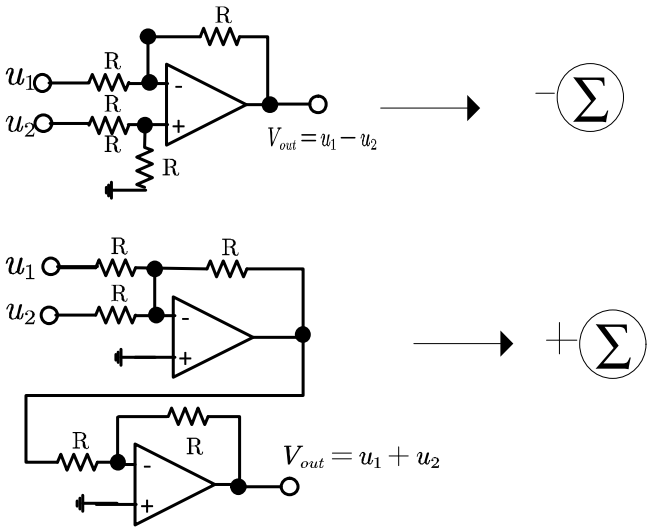

Fig. 14. Circuit implementation of operations about " + " and " − ".

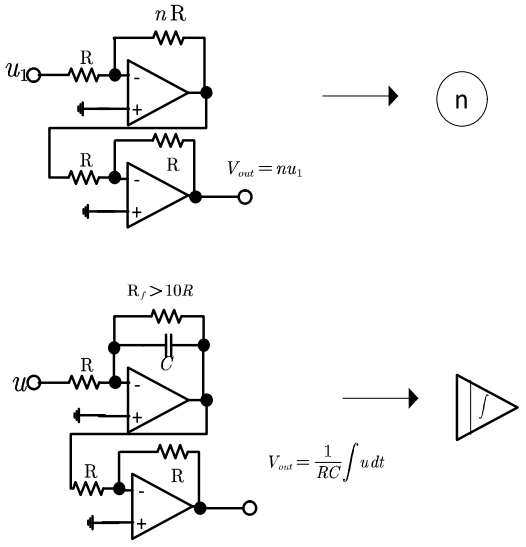

Fig. 15. Circuit implementation of multiplication operation $n\cdot$ and integral operation $\int \cdot$.

Op Amps, two diodes and two analog switches ("1/4 CD4066"). In Fig. 13, the discharge of capacitance is used to implement $\mu = \mu_0 e^{-2t}$ operation by selecting the appropriate values of resistances and capacitances. The arithmetic circuit of addition, subtraction, and integration is displayed in Figs. 14, 15. By combining the above basic circuits, we get the smoothing function $\hat{\theta}(x, \mu)$,i.e., $\hat{f}(x, \mu)$ and $\nabla_x \hat{\theta}(x, \mu)$, i.e., $\nabla_x \hat{f}(x, \mu)$.

Before giving a circuit of $w\left(A_j x - b_j\right)$ and $\nabla_x w\left(A_j x - b_j\right)$, several basic circuits of x^y with $y = \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{9}$ are displayed in Fig. 18 by using Op Amps and analog multipliers. (values $(1 > y > 0)$ except $\frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{9}$ can be achieved by using $y = \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{9}$ in suitable way with analog multipliers).

As can be see from Fig. 19, the circuit in the red box contains two resistors that are used to achieve the values of the $A_{ik}u_i$ with the values of two resistors of R_{ik} and $\frac{R_{ik}A_{ik}}{1-A_{ik}}$, respectively. Using the above circuit $(A_{ik}u_i)$ and the circuits in Figs. 11, 12, 14, 18, the term of $w\left(A_j x - b_j\right)$ can be implemented by circuit in Fig. 19.

In Fig. 20, a "1/4 LM339" (one voltage comparator) with $\pm 1V$ voltage values is used to implement the operation of sign (x), if $|x| > \frac{\mu}{2}$ (the red box in Fig. 20). By composing the elementary

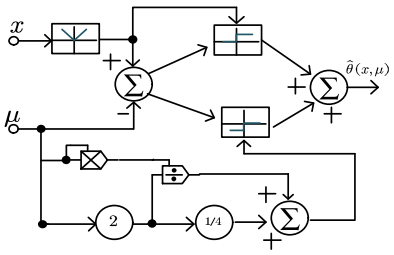

Fig. 16. Circuit implementation of term $\hat{\theta}(x, \mu)$.

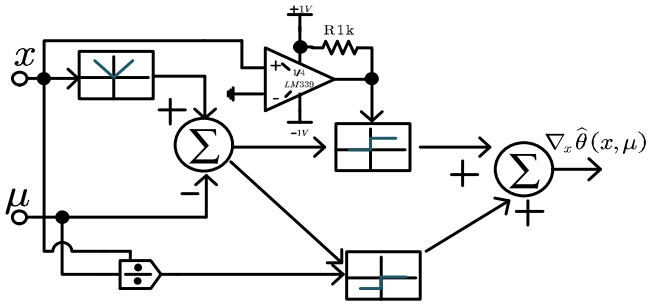

Fig. 17. Circuit implementation of term $\nabla_x \hat{\theta}(x, \mu)$.

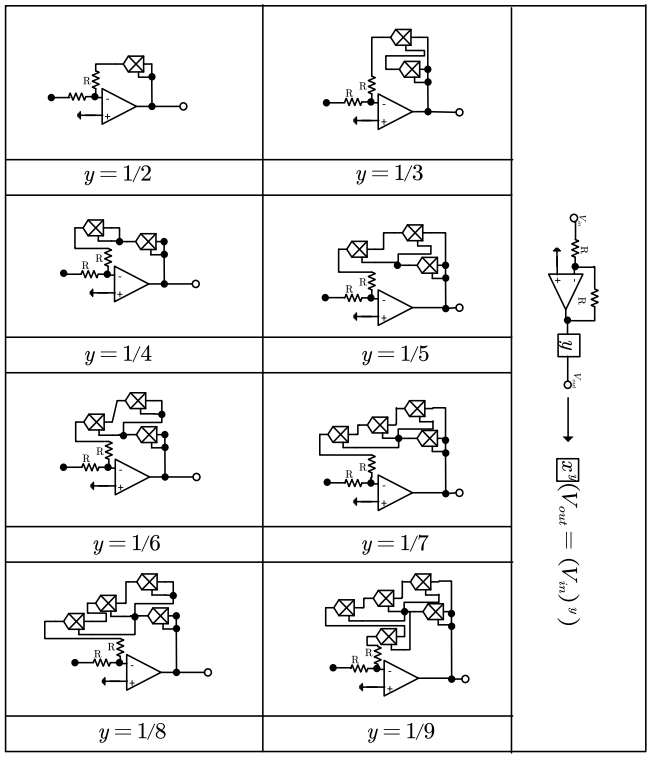

Fig. 18. Circuit implementation of term x^p.

circuits in Figs. 13–15, 18 and 19, the term of $\nabla_x w\left(A_j x - b_j\right)$ can be implemented by using the circuit in Fig. 20.

Fig. 21 displays the complete circuit of SNN (12). Though the expression of SNN looks complex, it can be realized based on the elementary circuits in Figs. 10–20 which means that it does not have expensive components in circuit implementation. The circuit in Fig. 21 is a feedback loop system which alters the voltage values until their equilibrium is reached which satisfies KKT condition. The detailed technique can refer correctto Chua,

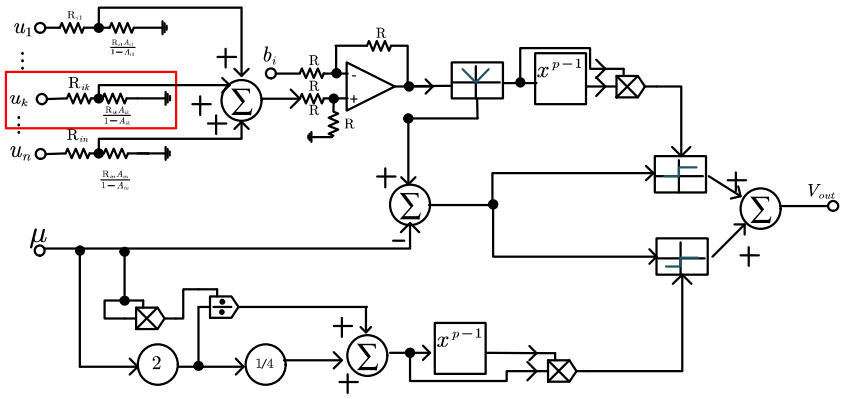

Fig. 19. Circuit implementation of term $w\left(A_j x - b_j\right)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

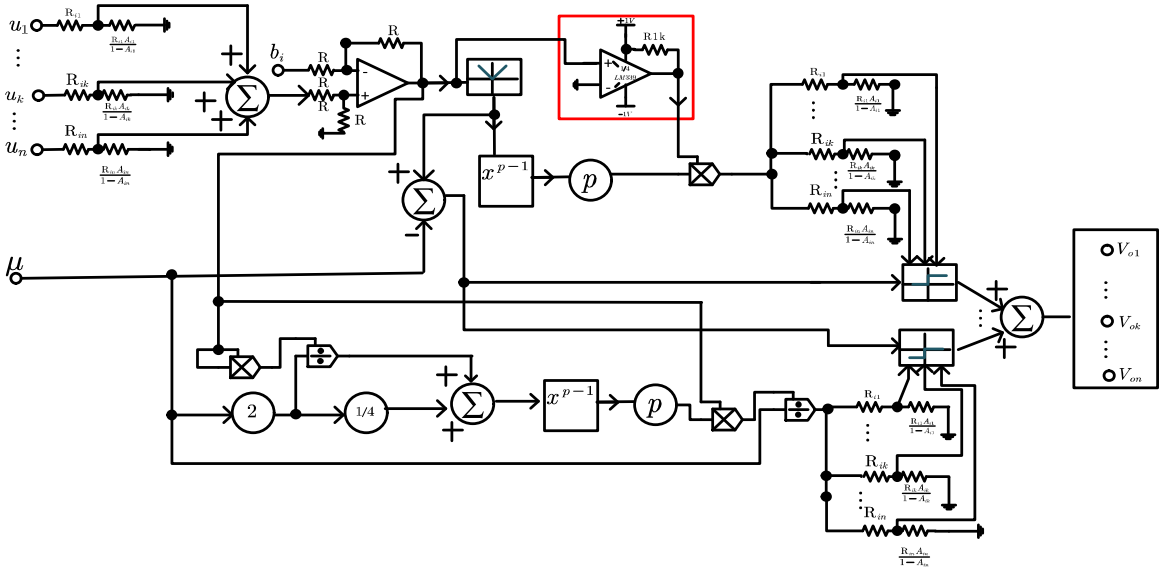

Fig. 20. Circuit implementation of term $\nabla_x w\left(A_j x - b_j\right)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Desoer, and Kuh (1987), Kennedy and Chua (1988) and Tank and Hopfield (1986).

7. Conclusions

Based on the smoothing approximation approach, this paper investigates a smoothing neural network for robust sparse signal recovery by solving minimization $l_1 - l_p$ $(2 \geq p \geq 1)$ problem in CS. Compared with existing optimization algorithms, such as BP-SEP, BPDN, LASSO, the proposed SNN has a high robustness in signal reconstruction problem in Gaussian noise and bit errors like corruption, but has a low robustness of L_p-ADM in signal reconstruction problem in Gaussian noise and bit errors like corruption. By using the theory of non-smooth analysis, Lyapunov method, we can prove that the proposed SNN can globally converge to an equilibrium point satisfied KKT condition under certain conditions. Simulation results on three examples problem have shown that the proposed recurrent neural network is feasible and very efficient. Future work is to further study minimization $l_1 - l_p$ $(1 \geq p \geq 0)$ i.e., an nonconvex, nonsmooth and non-Lipschitz problem and design an effective neural network to solve this problem and provides computational efficiency or the rate of convergence of the designed neural network algorithms. How to effectively select a better approximation method for different p is also the focus of our future work.

Acknowledgments

This work is supported in part by the Fundamental Research Funds for the Central Universities under Project XDJK2019B010, and also supported by Natural Science Foundation of China (Grant No: 61773320), and also supported by the Natural Science Foundation Project of Chongqing CSTC (Grant nos. cstc2018jcyjAX0583, cstc2018jcyjAX0810), and also supported by Research Foundation of Key laboratory of Machine Perception and Children's Intelligence Development funded by CQUE(16xjpt07), China, and also supported by the Foundation of Chongqing University of Education (Grant No.KY201702A). This publication was made possible by NPRP Grant No. NPRP 7-1482- 1-278 from the Qatar National Research Fund (a member of Qatar Foundation).

Appendix

Here, we present some definitions and properties of Clarke generalized gradient, and the convexity of $\hat{f}(x, \mu)$ and $\hat{g}(x, \mu)$, which are needed for the theoretical analysis in this paper. We refer the readers to see Aubin and Cellina (1984), Clarke (1983), Foucart and Lai (2009), Liu and Wang (2011), Rockafellar and Wets (1998), Studer and Baraniuk (2014) and Tsaig and Donoho (2008).

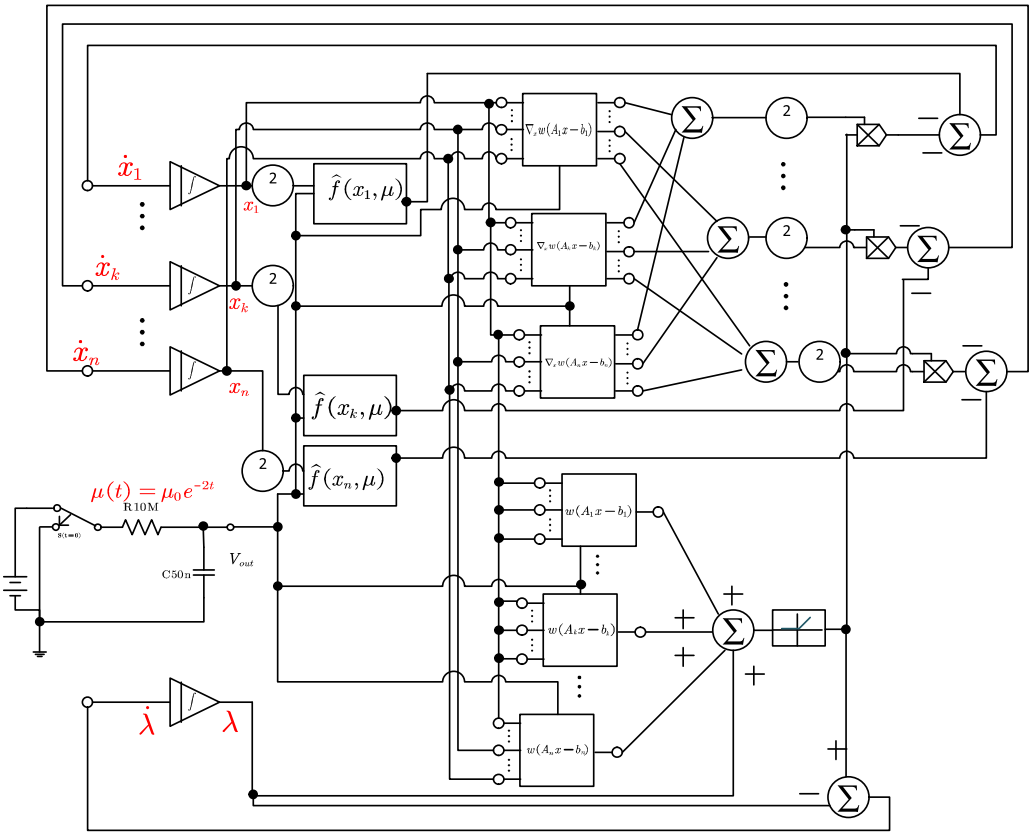

Fig. 21. Circuit implementation of SNN algorithm (12).

Definition 2: Let θ: $\mathbb{R}^n \to \mathbb{R}$ be function which satisfies locally Lipschitz, here θ is differentiable almost everywhere. Denote D_θ as the set of points which θ is differentiable. Then the Clarke generalized gradient is

$$\partial\theta(x) = co\left\{\lim_{x_k\to x, x_k\in D_\theta} \nabla\theta(x_k)\right\} \tag{51}$$

Proposition 1. *The smoothing function $\hat{f}(x,\mu)$ is convex with respect to x, also with respect to μ. It is easy to observe the convexity of $\hat{f}(x,\mu)$ at x according to the equality (4). By the definition of convex function, the $\hat{f}(x,\mu)$ is also a convex for μ.*

$$\hat{f}(x,\mu_1) - \hat{f}(x,\mu_2) - \nabla_{\mu_2}\hat{f}(x,\mu_2) = \begin{cases} 0 & , if\ |x| > \mu_1 \geq \mu_2,\ \ |x| > \mu_2 \geq \mu_1 \\ \frac{(x\pm\mu_1)^2}{2\mu_1} \geq 0 & , if\ \ \mu_1 > |x| > \mu_2 \\ \frac{(\mu_2(|x|-\mu_1)+|x|(\mu_2-\mu_1))(\mu_2-|x|)}{2\mu_2^2} \geq 0 & , if \mu_2 \geq |x| \geq \mu_1 \\ \frac{x^2(\mu_1-\mu_2)^2}{2\mu_1\mu_2^2} \geq 0 & , if\ |x| \leq \mu_1 \leq \mu_2, |x| \leq \mu_2 \leq \mu_1 \end{cases} \tag{52}$$

Then the smoothing function $\hat{g}(x,\mu)$ is also convex with respect to x, also for μ. It is easy to see the convexity of $\hat{f}(x,\mu)$ of x. We only show that the $\hat{g}(x,\mu)$ is a convex function of μ:

$$\nabla^2_\mu \hat{g}(x,\mu) = \begin{cases} 0, & if\ |x| > \mu_2 \\ \left\{p(p-1)\left(\frac{x^2}{2\mu}+\frac{\mu}{2}\right)^{p-2}\left(-\frac{x^2}{2\mu^2}+\frac{1}{2}\right)^2 \right. \\ \left. p\frac{x^2}{\mu^3}\left(\frac{x^2}{2\mu}+\frac{\mu}{2}\right)^{p-1} \geq 0\right\}, & if\ |x| \leq \mu_2 \end{cases} \tag{53}$$

So, the proof is completed.

References

Afonso, M. V., Bioucas-Dias, J. M., & Figueiredo, M. A. T. (2011). An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, *20*(3), 681–695.

Aubin, J. P., & Cellina, A. (1984). *Differential inclusion: set-valued maps and viability theory*. New York: Springer-Verlag.

Bellasi, D. E., & Benini, L. (2015). Energy-efficiency analysis of analog and digital compressive sensing in wireless sensors. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *62*(11), 2718–2729.

Bian, W., & Chen, X. (2012). Smoothing neural network for constrained non-lipschitz optimization with applications. *IEEE Transactions on Neural Networks Learning Systems*, *23*(3), 399–411.

Bian, W., & Chen, X. (2014). Neural network for nonsmooth nonconvex constrained minimization via smooth approximation. *IEEE Transactions on Neural Networks Learning Systems*, *25*(3), 545–556.

Candes, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathématique*, *346*(9), 589–592.

Candes, E. J., Romberg, J., & Tao, T. (2006a). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transaction on Information Theory*, *52*(2), 489–509.

Candes, E. J., Romberg, J. K., & Tao, T. (2006b). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, *59*(8), 1207–1223.

Chen, X. (2012). Smoothing methods for nonsmooth, nonconvex minimization. *Mathematical Programming*, 1–29.

Chen, X., Yu, Z., Hoyos, S., et al. (2011). A sub-Nyquist rate sampling receiver exploiting compressive sensing. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *58*(3), 507–520.

Cheng, L., Hou, Z., Lin, Y., et al. (2011). Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks. *IEEE Transactions on Neural Networks Learning Systems*, *21*(5), 714–726.

Cheng, L., Hou, Z., & Tan, M. (2009). A delayed projection neural network for solving linear variational inequalities. *IEEE Transactions on Neural Networks Learning Systems*, *20*(6), 915–925.

Cheng, L., Liu, W., Yang, C., Hou, Z., Huang, T., & Tan, M. (2018). A neural-network-based controller for piezoelectric-actuated stick–slip devices. *IEEE Transactions on Industrial Electronics*, *65*(3), 2598–2607.

Chua, L. O., Desoer, C. A., & Kuh, E. S. (1987). *Linear and nonlinear circuits*. New York, NY, USA: McGraw-Hill.

Clarke, F. H. (1983). *Optimization and nonsmooth analysis*. New York: Wiley.

Dong, Z. S., & Zhu, W. X. (2018). Homotopy methods based on L_0-norm for compressed sensing. *IEEE Transactions on Neural Networks Learning Systems*, *29*(42018), 1132–1146.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transaction on Information Theory*, *52*(4), 1289–1306.

Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2765–2781.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*, 1348–1360.

Feng, R., Leung, C. S., Constantinides, A. G., et al. (2017). Lagrange programming neural network for nondifferentiable optimization problems in sparse approximation. *IEEE Transactions on Neural Networks Learning Systems*, *28*(10), 2395–2407.

Figueiredo, M. A. T., Nowak, R. D., & Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, *1*(4), 586–597.

Foucart, S., & Lai, M. J. (2009). Sparsest solutions of underdetermined linear systems via l_q-minimization for $0 < q \leqslant 1$. *Applied and Computational Harmonic Analysis*, *26*(3), 395–407.

Guo, C., & Yang, Q. (2015). A neurodynamic optimization method for recovery of compressive sensed signals with globally converged solution approximating to l_0 minimization. *IEEE Transactions on Neural Networks Learning Systems*, *26*(7), 1363–1374.

Jiang, X., Zeng, W. J., Yasotharan, A., et al. (2014). Minimum dispersion beamforming for non-Gaussian signals. *IEEE Transactions on Signal Processing*, *62*(7), 1879–1893.

Karimi, H. R., & Gao, H. (2010). New delay-dependent exponential H_∞ synchronization for uncertain neural networks with mixed time delays. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *40*(1), 173–185.

Kennedy, M. P., & Chua, L. O. (1988). Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *35*(5), 554–562.

Koh, K., Kim, S. J., & Boyd, S. (2007). An interior-point method for large-scale l_1-regularized logistic regression. *Journal of Machine Learning Research*, 1519–1555.

Kulkarni, A., & Mohsenin, T. (2017). Low overhead architectures for OMP compressive sensing reconstruction algorithm. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, 1–13. http://dx.doi.org/10.1109/TCSI.2017.2648854, in press.

Leung, C. S., Sum, J., & Constantinides, A. G. (2014). Recurrent networks for compressive sampling. *Neurocomputing*, *129*, 298–305.

Li, Y., Cichocki, A., & Amari, S. I. (2004). Analysis of sparse representation and blind source separation. *Neural Computation*, *16*(6), 1193–1234.

Li, Y., Cichocki, A., & Amari, S. I. (2006). Blind estimation of channel parameters and source components for EEG signals: a sparse factorization approach. *IEEE Transactions on Neural Networks*, *17*(2), 419–431.

Li, Y. M., & Wei, D. (2016a). Delayed Lagrange neural network for sparse signal reconstruction under compressive sampling. *Optik*, *127*(18), 7077–7082.

Li, Y. M., & Wei, D. (2016b). Signal reconstruction of compressed sensing based on recurrent neural networks. *Optik*, *127*(10), 4473–4477.

Liu, Y., & Hu, J. (2016). A neural network for l_1-l_2 minimization based on scaled gradient projection: Application to compressed sensing. *Neurocomputing*, *173*, 988–993.

Liu, Q., & Wang, J. (2011). A one-layer recurrent neural network for constrained nonsmooth optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *41*(5), 1323–1333.

Liu, Q., & Wang, J. (2016). L_1-minimization algorithms for sparse signal reconstruction based on a projection neural network. *IEEE Transactions on Neural Networks Learning Systems*, *27*(3), 698–707.

Moghimi, F., Nasri, A., & Schober, R. (2009). L_p-norm spectrum sensing for cognitive radio networks impaired by non-Gaussian noise. In *GLOBECOM*, Honolulu, HI, USA, Nov. 30-Dec. 4 (pp. 1–6).

Rockafellar, R. T., & Wets, R. J. B. (1998). *Variational analysis*. Berlin, Germany: Springer-Verlag.

Studer, C., & Baraniuk, R. G. (2014). Stable restoration and separation of approximately sparse signals. *Applied and Computational Harmonic Analysis*, *37*(1), 12–35.

Tank, D., & Hopfield, J. J. (1986). Simple'neural'optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *33*(5), 533–541.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, *58*(1), 267–288.

Tomioka, R., & Sugiyama, M. (2009). Dual-augmented lagrangian method for efficient sparse reconstruction. *IEEE Signal Processing Letters*, *16*(12), 1067–1070.

Tsaig, Y., & Donoho, D. L. (2008). Fast solution of l_1-norm minimization problems when the solution maybe sparse. *IEEE Transaction on Information Theory*, *54*(11), 4789–4812.

Wagner, A., Wright, J., Ganesh, A., et al. (2012). Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(2), 372–386.

Wang, Y., Shen, H., Karimi, H. R., et al. (2018). Dissipativity-based fuzzy integral sliding mode control of continuous-time TS fuzzy systems. *IEEE Transactions on Fuzzy Systems*, *26*(3), 1164–1176.

Wang, D., & Zhang, Z. (2017). Generalized sparse recovery model and its neural dynamical optimization method for compressed sensing. *Circuits, Systems, and Signal Processing*, *36*(11), 4326–4353.

Wei, D. (2016). Novel convolution and correlation theorems for the fractional fourier transform. *Optik - International Journal for Light and Electron Optics*, *127*(7), 3669–3675.

Wei, D. (2018). New product and correlation theorems for the offset linear canonical transform and its applications. *Optik*, *164*, 243–253.

Wei, D., & Li, Y. M. (2016). Generalized sampling expansions with multiple sampling rates for lowpass and bandpass signals in the fractional Fourier transform domain. *IEEE Transactions on Signal Processing*, *64*(18), 4861–4874.

Wei, Y., Park, J. H., Karimi, H. R., et al. (2018). Improved stability and stabilization results for stochastic synchronization of continuous-time semi-Markovian jump neural networks with time-varying delay. *IEEE Transactions on Neural Networks Learning Systems*, *29*(6), 2488–2501.

Wen, F., Liu, P., Liu, Y., et al. (2017). Robust sparse recovery in impulsive noise via ℓ_p-ℓ_1 optimization. *IEEE Transactions on Signal Processing*, *65*(1), 105–118.

Xu, Z., Chang, X., Xu, F., et al. (2012). $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks Learning Systems*, *23*(7), 1013–1027.