Neurocomputing 129 (2014) 298–305

Contents lists available at ScienceDirect

Neurocomputing

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

Recurrent networks for compressive sampling

Chi-Sing Leung [a,*], John Sum [b], A.G. Constantinides [c]

[a] The Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, KLN, Hong Kong
[b] Institute of Technology Management, National Chung Hsing University, Taiwan
[c] Imperial College, UK

ARTICLE INFO

Article history:
Received 14 April 2013
Received in revised form
31 July 2013
Accepted 19 September 2013
Communicated by D. Wang
Available online 1 November 2013

Keywords:
Neural circuit
Stability

ABSTRACT

This paper develops two neural network models, based on Lagrange programming neural networks (LPNNs), for recovering sparse signals in compressive sampling. The first model is for the standard recovery of sparse signals. The second one is for the recovery of sparse signals from noisy observations. Their properties, including the optimality of the solutions and the convergence behavior of the networks, are analyzed. We show that for the first case, the network converges to the global minimum of the objective function. For the second case, the convergence is locally stable.

1. Introduction

Nonlinear constrained optimization problems have been studied over several decades [1–3]. Conventional ways for solving them are based on numerical methods [1,2]. As mentioned by many neural network pioneers [3–7], when realtime solutions are required, the neural circuit approach [3–5] is more effective. In the neural circuit approach, we do not solve them in a digital computer. Instead, we set up an associated neural circuit for the given constrained optimization problem. After the neural circuit settles down at one of the equilibrium points, the solution is obtained by measuring the neuron output voltages at this stable equilibrium point. So, one of the important issues is the stability of equilibrium points.

In the past two decades, many results of neural circuits were reported. For instance, Hopfield [4] investigated a neural circuit for solving quadratic optimization problems. In [5,8–10] a number of models were proposed to solve various nonlinear constrained optimization problems. The neural circuit approach is able to solve many engineering problems effectively. For example, it can be used for optimizing microcode [11]. Besides, it is able to search the maximum of a set of numbers [12–14]. In [15], the Lagrange programming neural network (LPNN) model was proposed to solve general nonlinear constrained optimization problems. For many years, many neural models for engineering problems were addressed. However, little attention has been paid to analog neural circuits for compressing sampling.

In compressing sampling [16–18], a sparse signal is sampled (measured) by a few random-like basis functions. The task of compressing sampling is to recover the sparse signal from the measurements. Compressive sampling can also be applied to a non-sparse signal by increasing the sparsity of the signal. It can be done by using some transform coding techniques. Conventional approaches for recovering sparse signals are based on the Newton's method [19–21].

As the neural circuit approach is a good alternative for nonlinear optimization, it is interesting to investigate the ways to apply the neural circuit approach for compressing sampling. This paper proposes two analog neural network models, based on LPNNs, for compressive sampling. One is for the standard recovery. Another one is for the recovery from noisy measurements. Since the norm-1 measure is not twice differentiable, it is difficult to construct a neural circuit. Hence this paper proposes an approximation for the objective function. With the approximation, the hyperbolic tangent function, which is a commonly used activation function in the neural network community, is involved. We use experiments to investigate how the hyperbolic tangent parameter affects the performance of the proposed models. Since the convergence and stability of neural circuits are important issues, this paper investigates the stability and optimality of our approaches. For the case of the standard recovery, we show that the proposed neural model converges to the global minimum of the objective function. For the case of the recovery from noisy measurements, we show that the neural model is locally stable.

This paper is organized as follows. In Section 2, the backgrounds of compressive sampling and LPNNs are reviewed. Section 3 formulates neural models for compressive sampling. The theoretical analysis on the proposed neural models is also presented.

* Corresponding author.
E-mail address: eeleungc@cityu.edu.hk (C.-S. Leung).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.09.028

Section 4 presents our simulation results. Section 5 concludes our results.

2. Background

2.1. Compressive sampling

In compressive sampling [16–18], we would like to find a sparse solution $\boldsymbol{x} \in \Re^n$ of an underdetermined system, given by

$$\boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{b} \in \Re^m$ is the observation vector, $\boldsymbol{\Phi} \in \Re^{m\times n}$ is the measurement matrix with a rank of m, $\boldsymbol{x} \in \Re^n$ is the unknown sparse vector to be recovered, and $m < n$. In a more precise way, the sparsest solution is defined as

$$\min \quad |\boldsymbol{x}|_0 \tag{2a}$$

$$\text{subject to} \quad \boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{x}. \tag{2b}$$

Unfortunately, problem (2) is NP-hard. Therefore, we usually replace the l_0-norm measure with the l_1-norm measure. The problem for recovering sparse signals becomes

$$\min \quad |\boldsymbol{x}|_1 \tag{3a}$$

$$\text{subject to} \quad \boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{x}. \tag{3b}$$

This problem is known as basis pursuit (BP) [17,19]. Let $\boldsymbol{\phi}_j$ be the j-th column of $\boldsymbol{\Phi}$. Define the mutual coherence [18] of $\boldsymbol{\Phi}$ as

$$\mu(\boldsymbol{\Phi}) = \max_{i\neq j} \frac{|\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j|}{|\boldsymbol{\phi}_i|_2 |\boldsymbol{\phi}_j|_2}. \tag{4}$$

If the cardinality of the true solution obeys $|\boldsymbol{x}|_0 < \frac{1}{2}(1+1/\mu(\boldsymbol{\Phi}))$, then the solution of (3) is exactly the same as that of (2). When the measurement matrix has independent Gaussian entries and the number m of measurements is greater than $2k\log(n/k)$, the BP [22] can reconstruct the sparse signal with high probability.

When there is measurement noise in $\boldsymbol{b}$, the sampling process becomes

$$\boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{\xi}, \tag{5}$$

where $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_m]^T$, and ξ_i's are independent identical random variables with zero mean and variance σ^2. In this case, our problem becomes

$$\min \quad |\boldsymbol{x}|_1 \tag{6a}$$

$$\text{subject to} \quad |\boldsymbol{b} - \boldsymbol{\Phi}\boldsymbol{x}|^2 \leq m\sigma^2. \tag{6b}$$

2.2. Lagrange programming neural networks

The LPNN approach aims at solving a general nonlinear constrained optimization problem, given by

$$\text{EP}: \quad \min \quad f(\boldsymbol{x}) \tag{7a}$$

$$\text{subject to} \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \tag{7b}$$

where $\boldsymbol{x} \in \Re^n$ is the state of system, $f: \Re^n \to \Re$ is the objective function, and $\boldsymbol{h}: \Re^n \to \Re^m$ $(m < n)$ describes the m equality constraints. The two functions f and $\boldsymbol{h}$ are assumed to be twice differentiable.

The LPNN approach first sets up a Lagrange function for EP, given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{x}), \tag{8}$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^T$ is the Lagrange multiplier vector. The LPNN approach then defines two kinds of neurons: variable neurons and Lagrange neurons. The variable neurons hold the state variable vector $\boldsymbol{x}$. The Lagrange neurons hold the Lagrange multiplier vector $\boldsymbol{\lambda}$.

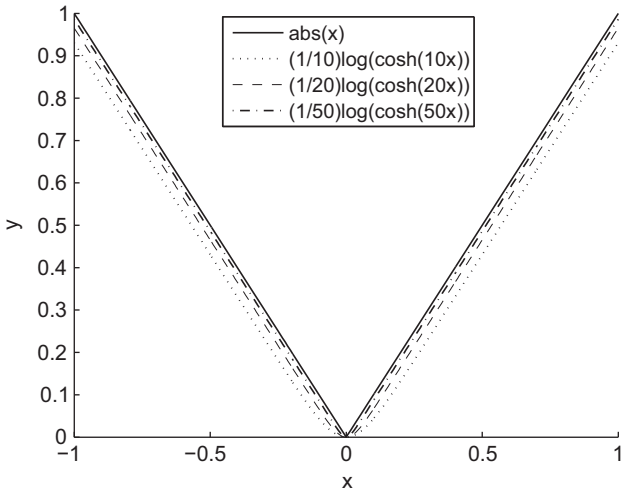

Fig. 1. $y = (1/a)\ln(\cosh(ax))$.

The LPNN approach defines two updating equations for those neurons, given by

$$\frac{1}{\varepsilon}\frac{d\boldsymbol{x}}{dt} = -\nabla_x \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \tag{9a}$$

$$\frac{1}{\varepsilon}\frac{d\boldsymbol{\lambda}}{dt} = \nabla_\lambda \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}), \tag{9b}$$

where ε is the time constant of the circuit. The time constant depends on the circuit resistance and the circuit capacitance. Without loss of generality, we set ε to 1. The variable neurons seek for a state with the minimum cost in a system while the Lagrange neurons are trying to constrain the system state of the system such that the system state falls into the feasible region. With (9), the network will settle down at a stable state [15] if the network satisfies some conditions.

3. LPNNs for compressive sampling

3.1. Sparse signal

From (3) and (7), one may suggest that we can set up a LPNN to solve problem (3). However, the absolute operator $|\cdot|$ is not differentiable at x_i=0. Hence, we need an approximation. In this paper, we use the following approximation:

$$|x_i| \approx \frac{\ln(\cosh(ax_i))}{a}, \tag{10}$$

where $a > 1$. Fig. 1 shows the shape of $(1/a)\ln(\cosh(ax))$. From the figure, the approximation[1] is quite accurate for a large a.

Thanks to the property of $(1/a)\ln(\cosh(ax))$, the hyperbolic tangent function is involved in the dynamic equations of the neural circuit. It should be noticed that the hyperbolic tangent function [23] is the commonly used activation in the neural network community. Also, comparators or amplifiers [24] are associated with the hyperbolic-tangent relation between input and output for differential bipolar pairs.

[1] In Section 4, we will use simulation examples to show the effectiveness of this approximation.

With the approximation, the problem becomes

$$\mathrm{EP}_{cs}: \ \min \ \ \frac{1}{a}\sum_{i}^{n}\ln(\cosh(ax_i)) \tag{11a}$$

$$\text{subject to} \quad \boldsymbol{b}-\boldsymbol{\Phi x}=\boldsymbol{0}. \tag{11b}$$

From (8), the Lagrange function for EP_{cs} is given by

$$\mathcal{L}_{cs}(\boldsymbol{x},\boldsymbol{\lambda})=\frac{1}{a}\sum_{i}^{n}\ln(\cosh(ax_i))+\boldsymbol{\lambda}^T(\boldsymbol{b}-\boldsymbol{\Phi x}). \tag{12}$$

Applying (9) on (12), we can obtain the neural dynamics for EP_{cs}, given by

$$\frac{d\boldsymbol{x}}{dt}=-\tanh(a\boldsymbol{x})+\boldsymbol{\Phi}^T\boldsymbol{\lambda} \tag{13a}$$

$$\frac{d\boldsymbol{\lambda}}{dt}=\boldsymbol{b}-\boldsymbol{\Phi x}. \tag{13b}$$

In the above equations, $\boldsymbol{\Phi}$ is the interconnection matrix between the variable neurons and Lagrange neurons. This kind of neural dynamics can be considered as a special form of bi-directional associative memories (BAMs) [25,26]. The information flow between two types of neurons is achieved through the interconnection matrix.

The modified constrained optimization problem (11) has several nice properties. First, the objective function is twice differentiable. Second, the objective function is convex and the constraints $\boldsymbol{b}-\boldsymbol{\Phi x}$ are affine. Hence any local optimum is the global optimum. Also, according to the Karush–Kuhn–Tucker (KKT) theorem, the necessary conditions are also sufficient for optimality. Based on these properties, we have the following theorem for EP_{cs}.

Theorem 1. *There is a unique solution for* EP_{cs}. *A state* $\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$ *is a unique solution of* EP_{cs}, *iff*

$$\nabla_x\mathcal{L}_{cs}(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)=\boldsymbol{0} \tag{14a}$$

$$\boldsymbol{b}-\boldsymbol{\Phi x}^*=\boldsymbol{0}. \tag{14b}$$

From (9), (14a) *can be rewritten as*

$$-\tanh(a\boldsymbol{x}^*)+\boldsymbol{\Phi}^T\boldsymbol{\lambda}^*=\boldsymbol{0}. \tag{15}$$

It should be noticed that Theorem 1 tells us the optimality of EP_{cs} only. It does not tell us that our proposed neural model, given by (13), converges to the optimal solution.

In the following, we will show that the proposed neural model, given by (13), leads to the optimal solution of EP_{cs}. That is, $\boldsymbol{x}(t)\to\boldsymbol{x}^*$ and $\boldsymbol{\lambda}(t)\to\boldsymbol{\lambda}^*$, as $t\to\infty$.

Theorem 2. *Given that the rank of* $\boldsymbol{\Phi}$ *is m, the neural model, given by* (13), *leads to the unique optimal solution* $\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$ *for* EP_{cs}. *That is,* $\boldsymbol{x}(t)\to\boldsymbol{x}^*$ *and* $\boldsymbol{\lambda}(t)\to\boldsymbol{\lambda}^*$, *as* $t\to\infty$.

Proof. Define a scalar function $V(\boldsymbol{x},\boldsymbol{\lambda})$, given by

$$V(\boldsymbol{x},\boldsymbol{\lambda})=\tfrac{1}{2}|\tanh(a\boldsymbol{x})-\boldsymbol{\Phi}^T\boldsymbol{\lambda}|^2+\tfrac{1}{2}|\boldsymbol{b}-\boldsymbol{\Phi x}|^2+\tfrac{1}{2}|\boldsymbol{x}-\boldsymbol{x}^*|^2+\tfrac{1}{2}|\boldsymbol{\lambda}-\boldsymbol{\lambda}^*|^2. \tag{16}$$

We will show that the scalar function (16) is a Lyapunov function for (13), and that the system is globally asymptotically stable. Clearly, $V(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)=0$ and $V(\boldsymbol{x},\boldsymbol{\lambda})>0$ for $\{\boldsymbol{x},\boldsymbol{\lambda}\}\neq\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$. Hence $V(\boldsymbol{x},\boldsymbol{\lambda})$ is low bounded. Also, $V(\boldsymbol{x},\boldsymbol{\lambda})\to\infty$, as $|\boldsymbol{x}|\to\infty$ and $|\boldsymbol{\lambda}|\to\infty$. This means, $V(\boldsymbol{x},\boldsymbol{\lambda})$ is radial unbounded.

Now we are going to show that $\dot{V}(\boldsymbol{x},\boldsymbol{\lambda})=dV(\boldsymbol{x},\boldsymbol{\lambda})/dt=0$ for $\{\boldsymbol{x},\boldsymbol{\lambda}\}=\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$, and $\dot{V}(\boldsymbol{x},\boldsymbol{\lambda})=dV(\boldsymbol{x},\boldsymbol{\lambda})/dt<0$ for $\{\boldsymbol{x},\boldsymbol{\lambda}\}\neq\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$.

The derivative of $V(\boldsymbol{x},\boldsymbol{\lambda})$ with respect to time is

$$\dot{V}(\boldsymbol{x},\boldsymbol{\lambda})=\frac{\partial V(\boldsymbol{x},\boldsymbol{\lambda})}{\partial\boldsymbol{x}}\frac{d\boldsymbol{x}}{dt}+\frac{\partial V(\boldsymbol{x},\boldsymbol{\lambda})}{\partial\boldsymbol{\lambda}}\frac{d\boldsymbol{\lambda}}{dt}. \tag{17}$$

Define

$$V(\boldsymbol{x},\boldsymbol{\lambda})=V_1(\boldsymbol{x},\boldsymbol{\lambda})+V_2(\boldsymbol{x},\boldsymbol{\lambda}) \tag{18}$$

$$V_1(\boldsymbol{x},\boldsymbol{\lambda})=\tfrac{1}{2}|\tanh(a\boldsymbol{x})-\boldsymbol{\Phi}^T\boldsymbol{\lambda}|^2+\tfrac{1}{2}|\boldsymbol{b}-\boldsymbol{\Phi x}|^2 \tag{19}$$

$$V_2(\boldsymbol{x},\boldsymbol{\lambda})=\tfrac{1}{2}|\boldsymbol{x}-\boldsymbol{x}^*|^2+\tfrac{1}{2}|\boldsymbol{\lambda}-\boldsymbol{\lambda}^*|^2. \tag{20}$$

It is easy to show that

$$\dot{V}_1(\boldsymbol{x},\boldsymbol{\lambda})=-(\tanh(a\boldsymbol{x})-\boldsymbol{\Phi}^T\boldsymbol{\lambda})^T\boldsymbol{A}(\tanh(a\boldsymbol{x})-\boldsymbol{\Phi}^T\boldsymbol{\lambda}), \tag{21}$$

where

$$\boldsymbol{A}=a\begin{pmatrix}1-\tanh^2(ax_1) & 0 & \cdots & 0\\ 0 & \ddots & \vdots & 0\\ \vdots & \cdots & \ddots & 0\\ 0 & \cdots & 0 & 1-\tanh^2(ax_n)\end{pmatrix}.$$

Note that $\boldsymbol{A}$ is positive definite. Also, we have

$$\dot{V}_2(\boldsymbol{x},\boldsymbol{\lambda})=-(\boldsymbol{x}-\boldsymbol{x}^*)^T(\tanh(a\boldsymbol{x})-\boldsymbol{\Phi}^T\boldsymbol{\lambda})+(\boldsymbol{\lambda}-\boldsymbol{\lambda}^*)^T(\boldsymbol{b}-\boldsymbol{\Phi x}). \tag{22}$$

From (14b) and (15), (22) becomes

$$\begin{aligned}\dot{V}_2(\boldsymbol{x},\boldsymbol{\lambda})&=-(\boldsymbol{x}-\boldsymbol{x}^*)^T(\tanh(a\boldsymbol{x})-\tanh(a\boldsymbol{x}^*))\\&\quad-(\boldsymbol{x}-\boldsymbol{x}^*)^T(\boldsymbol{\Phi}^T\boldsymbol{\lambda}^*-\boldsymbol{\Phi}^T\boldsymbol{\lambda})\\&\quad+(\boldsymbol{\lambda}-\boldsymbol{\lambda}^*)^T(\boldsymbol{b}-\boldsymbol{\Phi x}-\boldsymbol{b}+\boldsymbol{\Phi x}^*)\\&=-(\boldsymbol{x}-\boldsymbol{x}^*)^T(\tanh(a\boldsymbol{x})-\tanh(a\boldsymbol{x}^*)).\end{aligned} \tag{23}$$

For $\{\boldsymbol{x},\boldsymbol{\lambda}\}=\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$, we have $\dot{V}_1(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)=0$ and $\dot{V}_2(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)=0$. This means, $\dot{V}(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)=0$. For $\boldsymbol{x}=\boldsymbol{x}^*$ but $\boldsymbol{\lambda}\neq\boldsymbol{\lambda}^*$, we have $(\tanh(a\boldsymbol{x}^*)-\boldsymbol{\Phi}^T\boldsymbol{\lambda})\neq\boldsymbol{0}$ because the rank of $\boldsymbol{\Phi}$ is m.[2] Hence $\dot{V}_1(\boldsymbol{x}^*,\boldsymbol{\lambda})<0$ and $\dot{V}_2(\boldsymbol{x}^*,\boldsymbol{\lambda})=0$. This means, $\dot{V}(\boldsymbol{x}^*,\boldsymbol{\lambda})<0$. For $\boldsymbol{x}\neq\boldsymbol{x}^*$, we have $\dot{V}_1(\boldsymbol{x},\boldsymbol{\lambda})\leq 0$. Since $\tanh(\cdot)$ is a strictly monotonic function, $\dot{V}_2(\boldsymbol{x},\boldsymbol{\lambda})<0$. This means, $\dot{V}(\boldsymbol{x},\boldsymbol{\lambda})<0$. To sum up, we have $\dot{V}(\boldsymbol{x},\boldsymbol{\lambda})<0$ for $\{\boldsymbol{x},\boldsymbol{\lambda}\}\neq\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$, $\dot{V}(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)=0$, and $V(\boldsymbol{x},\boldsymbol{\lambda})$ is lower bounded at $\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$ and is radial unbounded. Hence $V(\boldsymbol{x},\boldsymbol{\lambda})$ is a Lyapunov function for (13). The network is globally asymptotically stable. It converges to the unique equilibrium point $\{\boldsymbol{x}^*,\boldsymbol{\lambda}^*\}$. The proof is complete. □

Theorem 2 tells us that the proposed neural model, given by (13), can find out the optimal solution of EP_{cs}. Note that the condition of Theorem 2 (rank of $\boldsymbol{\Phi}$ is m) usually holds in compressive sampling because the measurement vectors are usually independent.

We use a synthetic 1D sparse signal, shown in Fig. 2(a), to illustrate the convergence of the neural circuit. The signal length is 256. In this signal 15 data points have non-zero values. The connection matrix is a ± 1 random matrix. Fig. 2(b) shows the recovery from 60 measured values with no measurement noise. From the figure, the recovered signal is nearly the same as the original signal. Moreover, the network converges after 14 characteristic times, as shown in Fig. 2(c).

[2] Since the rank of $\boldsymbol{\Phi}$ is equal to m, there is a unique $\boldsymbol{\lambda}$, denoted as $\boldsymbol{\lambda}^*$, such that $(\tanh(a\boldsymbol{x}^*)-\boldsymbol{\Phi}^T\boldsymbol{\lambda}^*)=\boldsymbol{0}$.

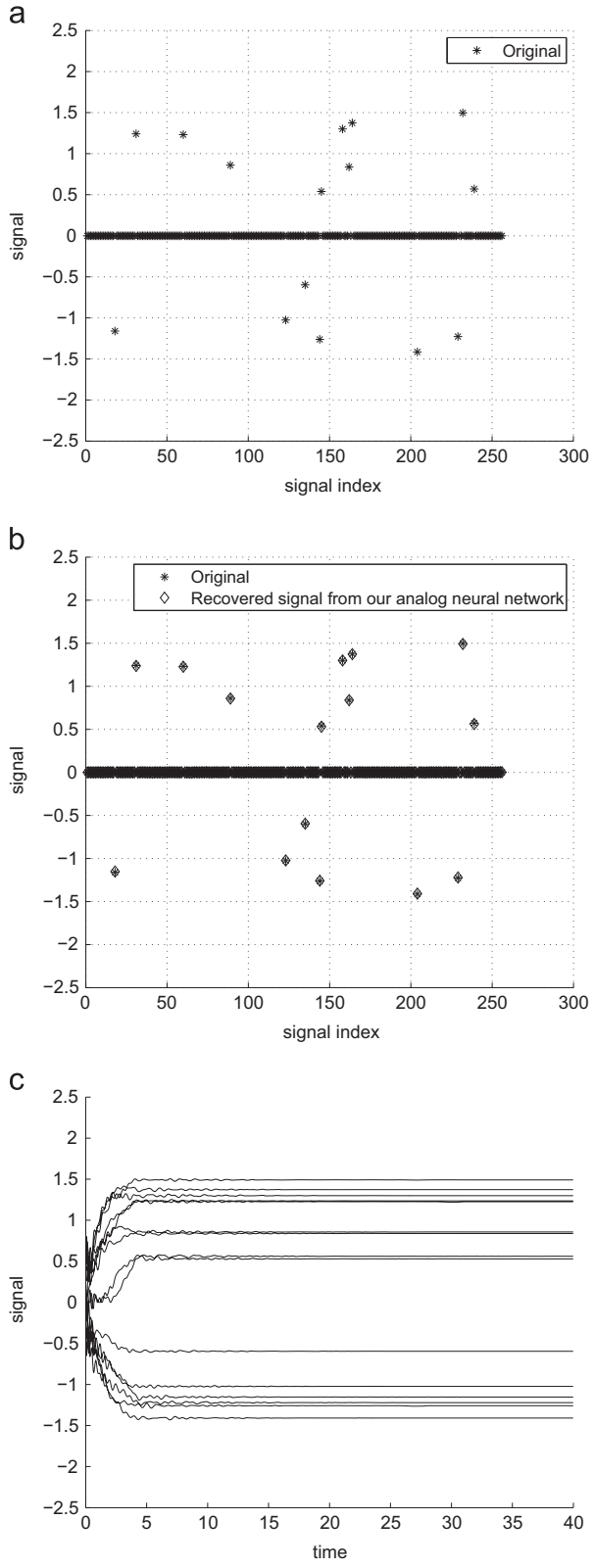

Fig. 2. The 1D sparse artificial signal recovery. (a) The sparse signal, (b) the recovered signal from the LPNN approach and (c) the dynamics of the recovered signal.

3.2. Noisy measurements

As mentioned before, when there is measurement noise in $\boldsymbol{b}$, the compressive sampling process can be represented as

$$\boldsymbol{b}=\boldsymbol{\Phi x}+\boldsymbol{\xi}, \tag{24}$$

where $\boldsymbol{\xi}=[\xi_1,\xi_2,\ldots,\xi_m]^T$, and ξ_i's are independently identically random variables with zero mean and variance σ^2. In this case, from (6) and (10), our problem becomes

$$\mathrm{EP}_{cn}:\ \min\ \frac{1}{a}\sum_i^n \ln(\cosh(ax_i)) \tag{25a}$$

$$\text{subject to}\quad |\boldsymbol{b}-\boldsymbol{\Phi x}|^2\le m\sigma^2. \tag{25b}$$

In our formulation, we assume that the signal power is greater than the noise power, i.e., $|\boldsymbol{b}|^2>m\sigma^2$. Since the objective function and the constraint are convex, any local optimum is a global optimum. According to the KKT theorem, we have the following theorem.

Theorem 3. *There is a unique solution for* EP_{cn}. *A state* $\{\boldsymbol{x}^*,\lambda^*\}$ *is a solution of* $\mathrm{EP}_{cn'}$, *iff*

$$\tanh(a\boldsymbol{x}^*)-\lambda^*\boldsymbol{\Phi}^T(\boldsymbol{b}-\boldsymbol{\Phi x}^*)=\boldsymbol{0} \tag{26}$$

$$|\boldsymbol{b}-\boldsymbol{\Phi x}^*|^2-m\sigma^2\le 0 \tag{27}$$

$$\lambda^*\ge 0 \tag{28}$$

$$\lambda^*(|\boldsymbol{b}-\boldsymbol{\Phi x}^*|^2-m\sigma^2)=0. \tag{29}$$

Note that in the noisy case λ^* *is a scalar.*

From Theorem 3 and the assumption $|\boldsymbol{b}|^2>m\sigma^2$, we have the following theorem.

Theorem 4. *Given* $|\boldsymbol{b}|^2>m\sigma^2$, *the optimization problem* EP_{cn} *is equivalent to*

$$\mathrm{EP}_{cn'}:\ \min\ \frac{1}{a}\sum_i^n \ln(\cosh(ax_i)) \tag{30a}$$

$$\text{subject to}\quad |\boldsymbol{b}-\boldsymbol{\Phi x}|^2-m\sigma^2=0 \tag{30b}$$

Proof. From (29), $\lambda^*>0$ is equivalent to $(|\boldsymbol{b}-\boldsymbol{\Phi x}^*|^2-m\sigma^2)=0$. From Theorem 3, EP_{cn} has a unique solution $\{\boldsymbol{x}^*,\lambda^*\}$ and $\lambda^*\ge 0$. At $\{\boldsymbol{x}^*,\lambda^*\}$, λ^* is either greater than zero or equal to zero. If $\lambda^*=0$, then from (26) $\boldsymbol{x}^*=\boldsymbol{0}$. This implies that from (27) $|\boldsymbol{b}|^2<m\sigma^2$. This contradicts our earlier assumption $|\boldsymbol{b}|^2>m\sigma^2$. Hence from (29) λ^* must be greater than zero and $|\boldsymbol{b}-\boldsymbol{\Phi x}^*|^2-m\sigma^2=0$. Therefore, the inequality in (25) can be removed, and the proof is complete. □

From Theorem 4, we consider the Lagrange function for (30), given by

$$\mathcal{L}_{cn'}(\boldsymbol{x},\boldsymbol{\lambda})=\frac{1}{a}\sum_i^n \ln(\cosh(ax_i))+\lambda(|\boldsymbol{b}-\boldsymbol{\Phi x}|^2-m\sigma^2). \tag{31}$$

Now, from (9) the neural model is given by

$$\frac{d\boldsymbol{x}}{dt}=-\tanh(a\boldsymbol{x})+2\lambda\boldsymbol{\Phi}^T(\boldsymbol{b}-\boldsymbol{\Phi x}) \tag{32}$$

$$\frac{d\lambda}{dt}=|\boldsymbol{b}-\boldsymbol{\Phi x}|^2-m\sigma^2. \tag{33}$$

From Theorem 3, we can obtain the stability property of (32) and (33), given by the following theorem.

Theorem 5. *Given* $|\boldsymbol{b}|^2>m\sigma^2$, *the unique solution* $\{\boldsymbol{x}^*,\lambda^*\}$ *of* EP_{cn} *and* $\mathrm{EP}_{cn'}$ *is an asymptotically stable point of the neural dynamics.*

Proof. The linearized neural dynamics around $\{\boldsymbol{x}^*,\lambda^*\}$ is given by

$$\begin{bmatrix}\frac{d\boldsymbol{x}}{dt}\\ \frac{d\lambda}{dt}\end{bmatrix}=-\begin{bmatrix}\boldsymbol{A}^*+2\lambda^*\boldsymbol{\Phi}^T\boldsymbol{\Phi} & 2\boldsymbol{\Phi}^T(\boldsymbol{b}-\boldsymbol{\Phi x}^*)\\ -2(\boldsymbol{b}-\boldsymbol{\Phi x}^*)^T\boldsymbol{\Phi} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{x}-\boldsymbol{x}^*\\ \lambda-\lambda^*\end{bmatrix} \tag{34}$$

where

$$\boldsymbol{A}^* = a\begin{pmatrix} 1-\tanh^2(ax_1^*) & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & 0 \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & 1-\tanh^2(ax_n^*) \end{pmatrix}.$$

Define

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{A}^* + 2\lambda^*\boldsymbol{\Phi}^T\boldsymbol{\Phi} & 2\boldsymbol{\Phi}^T(\boldsymbol{b}-\boldsymbol{\Phi}\boldsymbol{x}^*) \\ -2(\boldsymbol{b}-\boldsymbol{\Phi}\boldsymbol{x}^*)^T\boldsymbol{\Phi} & 0 \end{bmatrix}. \tag{35}$$

If the real part of each eigenvalue of $\boldsymbol{\Gamma}$ is strictly positive, then $\{\boldsymbol{x}^*, \lambda^*\}$ is an asymptotically stable point [27]. Denote $\tilde{\boldsymbol{x}}$ be the conjugate of $\boldsymbol{x}$. Let α be an eigenvalue of $\boldsymbol{\Gamma}$, and $(\chi^T, \tau)^T \neq (\boldsymbol{0}^T, 0)^T$ be the corresponding eigenvector. Clearly, if $(\chi^T, \tau)^T$ is an eigenvector of $\boldsymbol{\Gamma}$, it cannot be a zero vector.

Now we consider two cases, either $\chi = \boldsymbol{0}$ or $\chi \neq \boldsymbol{0}$. If $\chi = \boldsymbol{0}$, from the definition of eigenvector we have

$$\boldsymbol{\Gamma}\begin{bmatrix} \boldsymbol{0} \\ \tau \end{bmatrix} = \alpha\begin{bmatrix} \boldsymbol{0} \\ \tau \end{bmatrix}. \tag{36}$$

From the definition of $\boldsymbol{\Gamma}$ (see (35)), we also have

$$\boldsymbol{\Gamma}\begin{bmatrix} \boldsymbol{0} \\ \tau \end{bmatrix} = \begin{bmatrix} 2\tau\boldsymbol{\Phi}^T(\boldsymbol{b}-\boldsymbol{\Phi}\boldsymbol{x}^*) \\ 0 \end{bmatrix}. \tag{37}$$

For $|\boldsymbol{b}|^2 > m\sigma^2$, from (26), we have $\boldsymbol{x}^* \neq \boldsymbol{0}$, $\lambda^* \neq 0$ and $\boldsymbol{\Phi}^T(\boldsymbol{b}-\boldsymbol{\Phi}\boldsymbol{x}^*) \neq \boldsymbol{0}$. Therefore, $\tau = 0$. This contradicts the assumption of eigenvector. This means, $\chi \neq \boldsymbol{0}$.

Since $(\chi^T, \tau)^T$ is the eigenvector, we have

$$\mathrm{Re}\left\{[\tilde{\chi}^T\ \tilde{\tau}]\boldsymbol{\Gamma}\begin{bmatrix} \chi \\ \tau \end{bmatrix}\right\} = \mathrm{Re}(\alpha)(|\chi|^2 + |\tau|^2). \tag{38}$$

From the definition of $\boldsymbol{\Gamma}$, we also have

$$\mathrm{Re}\left\{[\tilde{\chi}^T\ \tilde{\tau}]\boldsymbol{\Gamma}\begin{bmatrix} \chi \\ \tau \end{bmatrix}\right\} = \mathrm{Re}\{\tilde{\chi}^T(\boldsymbol{A}^* + 2\lambda^*\boldsymbol{\Phi}^T\boldsymbol{\Phi})\chi\}. \tag{39}$$

This means, we have

$$\mathrm{Re}\{\tilde{\chi}^T(\boldsymbol{A}^* + 2\lambda^*\boldsymbol{\Phi}^T\boldsymbol{\Phi})\chi\} = \mathrm{Re}(\alpha)(|\chi|^2 + |\tau|^2). \tag{40}$$

Since $\boldsymbol{A}^*$ is positive definite and $\lambda^* > 0$, the left hand side must be greater than zero. This means $\mathrm{Re}(\alpha) > 0$. The proof is complete. □

We use the previous synthetic sparse signal (Fig. 2) to illustrate the convergence behavior. The measurement noise variance is 0.01. Fig. 3(a) shows the recovery from 60 measured values. The recovered signal has converged after 5 characteristic times, as shown in Fig. 3(b).

3.3. LPNNs for recovering non-sparse signal

Compressing sampling can be extended to handle non-sparse signals [28]. Let $\boldsymbol{z}$ be a non-sparse signal, and let $\boldsymbol{\Psi}$ be invertible transform which can transform a non-sparse signal to a sparse one. In this case, the measurement signal is given by $\boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{z}$. Clearly, the compressing sensing problem for non-sparse signals [28] becomes

$$\min \quad |\boldsymbol{\Psi}\boldsymbol{z}|_1 \tag{41a}$$

$$\text{subject to} \quad |\boldsymbol{b}-\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{z}|^2 \leq \varepsilon \tag{41b}$$

where $\varepsilon > 0$ is the error level in the residual. By considering $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{z}$, we can use the neural model presented in Section 3.2 to recover non-sparse signals.

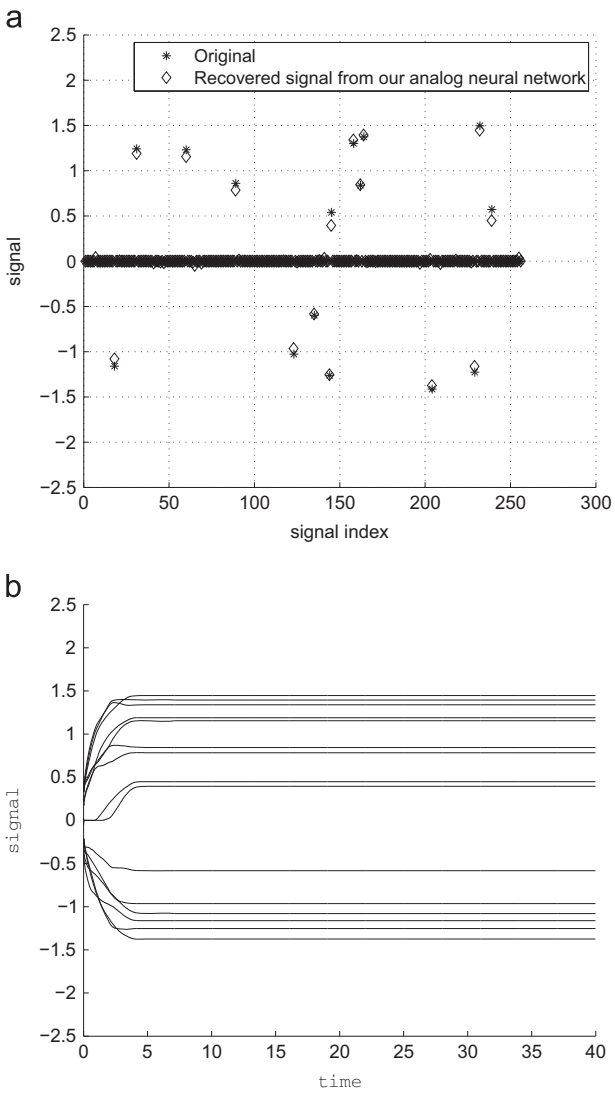

Fig. 3. The 1D sparse artificial signal recovery from noisy measurements. (a) The recovered signal from our approach where the measurements contain noise and the noise variance is 0.01. (b) The dynamics of the recovered signal.

3.4. Activation function in the dynamics

In (13) and (32), the hyperbolic tangent function is involved. In hardware implementation, comparators or amplifiers [24] are associated with the hyperbolic-tangent relation between input and output for differential bipolar pairs. Besides, the hyperbolic function [23] is the commonly used activation.

Instead of using the hyperbolic tangent function as the activation function, we can use any activation function $\varphi(x)$ with the following properties:

$$\lim_{x\to-\infty}\varphi(x) = -1, \tag{42a}$$

$$\lim_{x\to\infty}\varphi(x) = 1, \tag{42b}$$

$$\varphi(0) = 0 \tag{42c}$$

$$\lim_{x\to\pm\infty}\frac{d\varphi(x)}{dx} = 0, \tag{42d}$$

$$\frac{d\varphi(x)}{dx} > 0. \tag{42e}$$

With those properties, we can also obtain the same theoretical results about the stability and optimality. It is because the proofs of Theorems 1–5 are based on those properties. Hence, we can cascade two low gain neurons to form a high gain neuron, given by

$$\varphi(x) = \tanh(a \tanh(ax)). \tag{43}$$

4. Simulations

4.1. Recovery for sparse signals

In our approach, we use hyperbolic tangent for approximation. This section tests our neural model with different hyperbolic tangent parameter values $a=\{10,20,50\}$. The cascade approach, with activation function $\varphi(x) = \tanh(20 \tanh(20x))$ is also considered. As a comparison, we use the standard BP [21] from l_1-Magic to recover sparse signals.

The example is a standard data configuration used in many papers [29,21]. The signal length is 512 and 15 data points have non-zero values. We randomly generate 100 signals in our experiment. One of the 100 sparse signals is shown in Fig. 4. The measurement matrix $\mathbf{\Phi}$ is a ± 1 random matrix. We vary the number of measurements from 60 to 120. We repeat our experiment 100 times with different random matrices $\mathbf{\Phi}$ and different sparse signals $\boldsymbol{x}$.

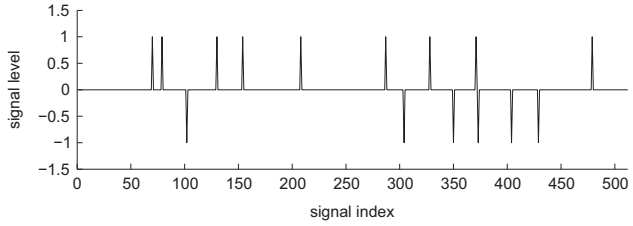

Fig. 4. The sparse signal. One of the 100 sparse signals in Sections 4.1 and 4.2.

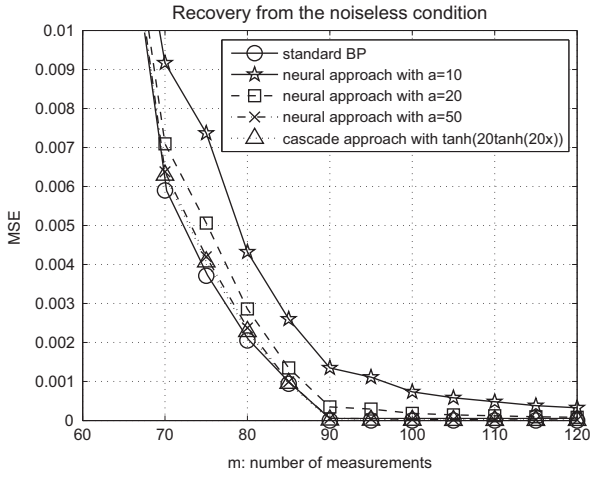

Fig. 5. Average MSE of the recovery signals. The signal length is 512 and 15 data points have non-zero values. We repeat our experiment 100 times with different random matrices and different sparse signals.

Fig. 5 shows the average mean square error (MSE) values of recovery signals. From the figure, we can observe that for the standard BP, we need around 90 measurement signals to recover the sparse signal. For our method, when the value of the hyperbolic tangent parameter is too small, such as 10 or 20, there is noticeable degradation in MSE values. When the hyperbolic tangent parameter a is equal to 50, there is no noticeable difference in MSE values between the standard BP method and our neural approach. Also, when we use the cascade approach in the activation function ($\varphi(x) = \tanh(20 \tanh(20x))$), there is no noticeable difference in MSE values between the standard BP method and the cascade activation function approach.

4.2. Recovery from noisy measurements

This section tests our neural model for recovering sparse signals from noisy measurements. The simulation setting is similar to that of Section 4.1. We add Gaussian noise in the measurement signals. The experiment considers three signal to noise (SNR) levels, {45 dB, 31 dB, 25 dB}, in the measurement signal. For our neural approach, three different hyperbolic tangent parameter values, $a=\{10,20,50\}$, are considered. In the simulation, we also consider the cascade activation function approach with $\varphi(x) = \tanh(20 \tanh(20x))$.

As a comparison, we use the "Min-l_1 with quadratic constraints" (ML_1QC) algorithm from l_1-Magic [21] to recover sparse signals. We repeat our experiment 100 times with different random matrices $\mathbf{\Phi}$ and different sparse signals $\boldsymbol{x}$. Fig. 6 shows the average MSE values of recovery signals. From the figure, we can observe that for the ML_1QC algorithm, we need around 90 measurement signals to recover the sparse signal. For our method, when the value of the hyperbolic tangent parameter is too small, such as 10 or 20, there is noticeable degradation. When the hyperbolic tangent parameter a is equal to 50, there is no noticeable difference in MSE values between the ML_1QC method and our approach. Also, when we use the cascade approach in the activation function ($\varphi(x) = \tanh(20 \tanh(20x))$), there is no noticeable difference in MSE values between the ML_1QC method and the cascade activation function approach.

4.3. Recovery for non-sparse signal

For recovering non-sparse signals, we can consider the situation of data compression [28]. We use the algorithm presented in Section 3.3 to recover non-sparse signals. We consider a 1D signal which is extracted from one row of image "CameraMan". The

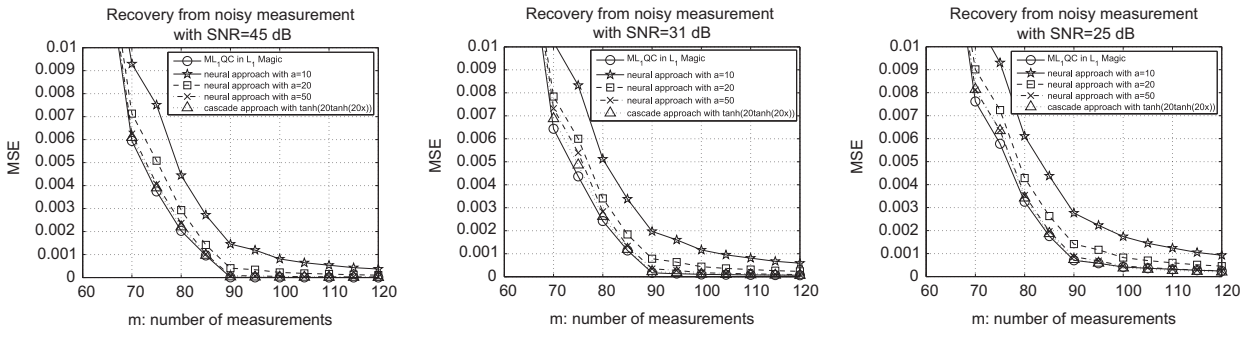

Fig. 6. Average MSE of the recovery signals from noisy measurements. At each number of measurements and at each measurement noise levels, we repeat our experiment 100 times with different random matrices and different sparse signals.

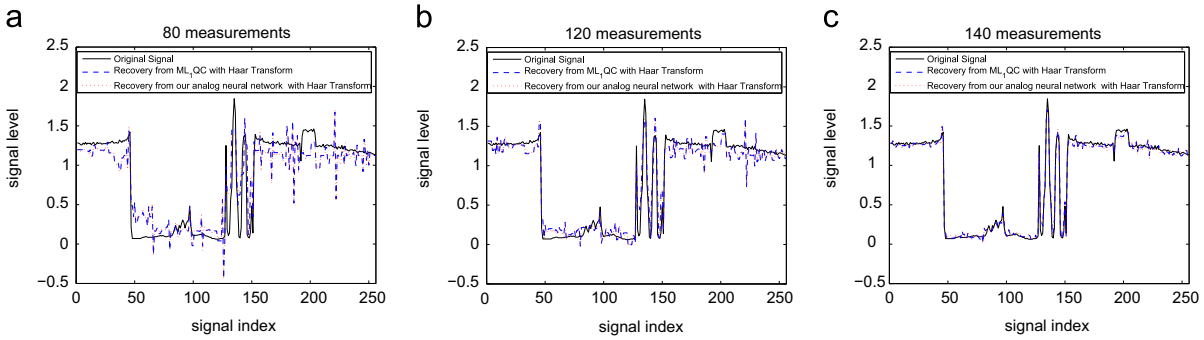

Fig. 7. Signal recovery from the non-sparse signal.

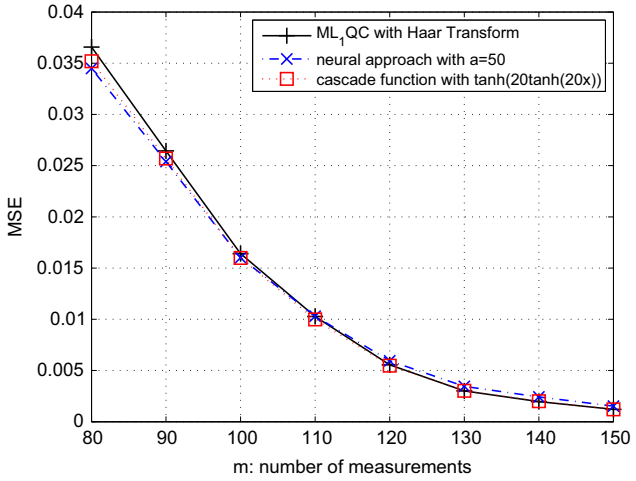

Fig. 8. MSE of signal recovery for non-sparse signals. We repeat our experiment 100 times with different random matrices Φ.

transform used is the Haar transform. As a comparison, we also consider the ML_1QC algorithm. We vary the number of measurements in our experiment. The reconstruction signals of using different number of measurements are shown in Fig. 7. From the figure, when more measurements are considered, we get a better reconstruction. Also, the quality of using our neural method is similar to that of using the ML_1QC algorithm. We repeat our experiment 100 times with different random matrices. Fig. 8 shows that our neural approach and the ML_1QC algorithm produce a similar MSE performance.

5. Conclusion

We formulated two analog neural models for signal recovery in compressive sampling. We proposed neural dynamics to handle two cases, including the standard recovery and the recovery from noisy measurements. We not only proposed the neural models but also investigated the stability of our neural models. We showed that for the case of the standard recovery, our proposed neural dynamics converge to the unique global minimum solution. For the case of recovery from noisy measurements, we showed that our proposed neural model is locally stable. With our stability analysis, the neural dynamics has a great potential application in compressive sampling. Experimental results showed that there are no significant differences between our analog approach and the traditional numerical methods. One of the future directions is to modify the dynamics for the noisy situation so that the neural dynamics are globally stable.

Acknowledgment

The work presented in this paper is supported by a research grant (CityU 115612) from the Research Grants Council of the Government of the Hong Kong Special Administrative Region.

References

[1] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear Programming: Theory and Algorithms, 2nd ed., Wiley, New York, 1993.
[2] A. Ruszczynski, Nonlinear Optimization, , 2006.
[3] A. Cichocki, R. Unbehauen, Neural Networks for Optimization and Signal Processing, Wiley, London, UK, 1993.
[4] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. U.S.A. 79 (1982) 2554–2558.
[5] L.O. Chua, G.N. Lin, Nonlinear programming without computation, IEEE Trans. Circuits Syst. 31 (1984) 182–188.
[6] Y. Xia, G. Feng, J. Wang, A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints, IEEE Trans. Neural Networks 19 (8) (2008) 1340–1353.
[7] Q. Liu, J. Wang, A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming, IEEE Trans. Neural Networks 19 (4) (2008) 558–570.
[8] C. Dang, Y. Leung, X. Bao Gao, K. Zhou Chen, Neural networks for nonlinear and mixed complementarity problems and their applications, Neural Networks 17 (2004) 271–283.
[9] X. Hu, J. Wang, Solving pseudomonotone variational inequalities and pseudo-convex optimization problems using the projection neural network, IEEE Trans. Neural Networks 17 (6) (2006) 1487–1499, http://dx.doi.org/10.1109/TNN.2006.879774.
[10] X. Hu, J. Wang, A recurrent neural network for solving a class of general variational inequalities, IEEE Trans. Syst. Man Cybern., Part B: Cybern. 37 (3) (2007) 528–539.
[11] S. Bharitkar, K. Tsuchiya, Y. Takefuji, Microcode optimization with neural networks, IEEE Trans. Neural Networks 10 (3) (1999) 698–703.
[12] J. Sum, C.-S. Leung, P. Tam, G. Young, W. Kan, L.-W. Chan, Analysis for a class of winner-take-all model, IEEE Trans. Neural Networks 10 (1) (1999) 64–71.
[13] J. Wang, Analysis and design of a k-winners-take-all model with a single state variable and the heaviside step activation function, IEEE Trans. Neural Networks 21 (9) (2010) 1496–1506, http://dx.doi.org/10.1109/TNN.2010.2052631.
[14] Y. Xiao, Y. Liu, C.S. Leung, J. Sum, K. Ho, Analysis on the convergence time of dual neural network-based KWTA, IEEE Trans. Neural Networks Learn. Syst. 23 (4) (2012) 676–682.
[15] S. Zhang, A.G. Constantinidies, Lagrange programming neural networks, IEEE Trans. Circuits Syst. II 39 (1992) 441–452.
[16] D. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition, IEEE Trans. Inf. Theory 47 (7) (1999) 2845–2862.
[17] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 43 (1) (2001) 129–159.
[18] D. Donoho, M. Elad, Optimally sparse representation in general dictionaries via l1 minimization, Proc. Natl. Acad. Sci. U.S.A. 100 (5) (2003) 2197–2202.
[19] E.J. Candès, M.B. Wakin, An introduction to compressive sampling, IEEE Signal Process. 25 (2008) 21–30.
[20] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodological) 58 (1996) 267–288.
[21] E. Candes, J. Romberg, L1 Magic: Recovery of Sparse Signals via Convex Programming. URL ⟨http://users.ece.gatech.edu/~justin/l1magic/⟩.
[22] D.L. Donoho, Jared Tanner, Counting faces of randomly-projected polytopes when the projection radically lowers dimension, J. AMS (2009) 1–53.
[23] A. Annema, Hardware realisation of a neuron transfer function and its derivative, Electron. Lett. 30 (7) (1994) 576–577.
[24] A. Moscovici, High Speed A/D Converters: Understanding Data Converters Through SPICE, Kluwer Academic Publishers, 2001.

[25] B. Kosko, Bidirectional associative memories, IEEE Trans. Syst. Man Cybern. 18 (1) (1988) 49–60 http://dx.doi.org/10.1109/21.87054.
[26] C.S. Leung, L.W. Chan, E. Lai, Stability and statistical properties of second-order bidirectional associative memory, IEEE Trans. Neural Networks 8 (2) (1997) 267–277.
[27] J.-J.E. Slotine, W. Li, Applied Nonlinear Control, Prentice Hall, 1991.
[28] Y. Tsaig, D.L. Donoho, Extensions of compressed sensing, Signal Process. 86 (3) (2006) 549–571.
[29] S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing, IEEE Trans. Signal Process. 56 (6) (2007) 2346–2356, http://dx.doi.org/10.1109/TSP.2007.914345.

Chi-Sing Leung received the B.Sci. degree in electronics, the M.Phil. degree in information engineering, and the PhD. degree in computer science from the Chinese University of Hong Kong in 1989, 1991, and 1995, respectively. He is currently an Associate Professor in the Department of Electronic Engineering, City University of Hong Kong. His research interests include neural computing, data mining, and computer graphics. In 2005, he received the 2005 IEEE Transactions on Multimedia Prize Paper Award for his paper titled, "The Plenoptic Illumination Function" published in 2002. He was a member of Organizing Committee of ICONIP2006. He was the Program Chair of ICONIP2009 and ICONIP2012. He is/was the guest editors of several journals, including, Neurocomputing, and Neural Computing and Applications. He is a governing board member of the Asian Pacific Neural Network Assembly (APNNA) and Vice President of APNNA.

John Sum received the B.Eng. in Electronic Engineering from the Hong Kong Polytechnic University in 1992, M.Phil. and Ph.D. in CSE from the Chinese University of Hong Kong in 1995 and 1998. John spent 6 years teaching in several universities in Hong Kong, including the Hong Kong Baptist University, the Open University of Hong Kong and the Hong Kong Polytechnic University. In 2005, John moved to Taiwan and started to teach in Chung Shan Medical University. Currently, he is an associate professor in the Institute of E-Commerce, the National Chung Hsing University, Taichung, ROC. His research interests include neural computation, mobile sensor networks and scale-free network. John Sum is a senior member of IEEE and an associate editor of the International Journal of Computers and Applications.

A.G. Constantinides is the Professor of Signal and the head of the Communications and Signal Processing Group of the Department of Electrical and Electronic Engineering. He has been actively involved in research in various aspects of digital filter design, digital signal processing, and communications for more than 30 years. Professor Constantinides' research spans a wide range of Digital Signal Processing and Communications, both from the theoretical as well as the practical points of view. His recent work has been directed toward the demanding signal processing problems arising from the various areas of mobile telecommunication. This work is supported by research grants and contracts from various government and industrial organisations.

Professor Constantinides has published several books and over 250 papers in learned journals in the area of Digital Signal Processing and its applications. He has served as the First President of the European Association for Signal Processing (EURASIP) and has contributed in this capacity to the establishment of the European Journal for Signal Processing. He has been on, and is currently serving as, a member of many technical program committees of the IEEE, the IEE and other international conferences. He has organised the first ever international series of meetings on Digital Signal Processing, in London initially in 1967, and in Florence (with Vito Cappellini) since 1972. In 1985 he was awarded the Honour of Chevalier, Palmes Academiques, by the French government, and in 1996, the promotion to Officier, Palmes Academiques. He holds honorary doctorates from European and Far Eastern Universities, several Visiting Professorships, Distinguished Lectureships, Fellowships and other honours around the world.