A novel one-layer recurrent neural network for the l_1-regularized least square problem

Mohammadi, Majid; Tan, Yao Hua; Hofman, Wout; Mousavi, S. Hamid

DOI
10.1016/j.neucom.2018.07.007

Publication date
2018

Document Version
Final published version

Published in
Neurocomputing

Neurocomputing 315 (2018) 135–144

Contents lists available at ScienceDirect

Neurocomputing

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

A novel one-layer recurrent neural network for the l_1-regularized least square problem

Majid Mohammadi[a,*], Yao-Hua Tan[a], Wout Hofman[b], S. Hamid Mousavi[c]

[a] Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands
[b] The Netherlands Institute of Applied Technology (TNO)
[c] Department of Medical Physics and Acoustics and Cluster of Excellence Hearing4all, Carl von Ossietzky University of Oldenburg, Germany

ARTICLE INFO

Article history:
Received 18 July 2017
Revised 12 May 2018
Accepted 4 July 2018
Available online 10 July 2018

Communicated by Dr Ding Wang

Keywords:
Least squares
l_1-regularization
Recurrent neural network
Convex
Lyapunov
Total variation

ABSTRACT

The l_1-regularized least square problem has been considered in diverse fields. However, finding its solution is exacting as its objective function is not differentiable. In this paper, we propose a new one-layer neural network to find the optimal solution of the l_1-regularized least squares problem. To solve the problem, we first convert it into a smooth quadratic minimization by splitting the desired variable into its positive and negative parts. Accordingly, a novel neural network is proposed to solve the resulting problem, which is guaranteed to converge to the solution of the problem. Furthermore, the rate of the convergence is dependent on a scaling parameter, not to the size of datasets. The proposed neural network is further adjusted to encompass the total variation regularization. Extensive experiments on the l_1 and total variation regularized problems illustrate the reasonable performance of the proposed neural network.

1. Introduction

The l_1-regularized least squares, or the lasso [1], has received a considerable amount of attention over the last decade and much research in recent years has focused on solving its non-smooth convex optimization problem

$$\min_x \quad \frac{1}{2}\|y - Ax\|_2^2 + \lambda\|x\|_1 \tag{1}$$

where $x \in \mathbb{R}^l, y \in \mathbb{R}^n, A$ is an $n \times l$ matrix consisting of l data points, λ is a non-negative parameter, $\|v\|_2$ indicates the Euclidean norm, and $\|v\|_1 = \Sigma|v_i|$ is the l_1-norm of v, which encourages the small components of x to be zero.

The lasso has a broad range of applications, such as signal reconstruction [2], curve fitting and classification [3], subspace clustering [4,5], sparse coding [6,7], and robot control [8], to name just a few. In these applications, it is critical to solve the minimization (1) efficiently. Therefore, myriad methods have been developed for solving (1) more quickly and effectively [9–12].

One promising way to find the optimum of the minimization (1) is to utilize the recurrent neural network. One of the main advantages of such an approach is that the structure of RNNs can be implemented using very-large-scale integration (VLSI) and optical technologies. Furthermore, it is well-known that neural networks have the ability to process real-time applications. Hence, when there are demands on real-time processing, it is necessary and desirable to employ parallel and distributed approaches, like neural networks. Despite having such unique merits, solving the minimization (1) via RNNs is thoroughly neglected (with the exception of the RNNs for general non-smooth problems). And, it is the principal incentive to develop a novel recurrent neural network especially tailored for the lasso.

The tremendous challenge of solving the minimization (1) is its non-differentiability due to its l_1-regularization. There are two options to put forward the neural network by circumventing the non-differentiability of the lasso. The first approach is to take advantage of the dual problem of the minimization (1). This is the modus operandi of various methods in the recent literature [10,11,13]. The interior-point method is arguably the most famous technique used to solve the dual problem. Contrary to conventional interior-points methods, it is claimed that this technique is suitable for large-scale problems; a problem with millions of variables is soluble in several minutes on an ordinary PC. However, the main difficulty in solving the dual problem is finding the optimal solution of the primal problem, e.g. x in the minimization (1), from the dual solution. The calculation of the primal solution x from the dual variable has usually enmeshed the computation

* Corresponding author.
E-mail address: m.mohammadi@tudelft.nl (M. Mohammadi).

https://doi.org/10.1016/j.neucom.2018.07.007
0925-2312/

$(A^TA)^{-1}$. Mathematically speaking, such an inverse does not theoretically exist for all matrices A. On top of that, the inverse calculation is both time- and memory-consuming for large-scale problems. Therefore, this approach is not taken into account.

Another approach to solve the minimization (1) is to convert it into a smooth problem by splitting the variable x into its positive and negative parts. The resultant smooth problem can be readily solved using gradient-based methods. The gradient projection for sparse reconstruction (GPSR) [9] solves the smooth problem and is of immense popularity among other methods. Further studies on the gradient projection concentrated on accelerating the convergence [14,15].

In this article, we use the second approach to come up with a neural network in order to avoid the calculation of the inverse matrix. However, splitting the variable into its positive and negative parts results in dimension escalation of the consequent smooth problem. We further investigate whether the dimension increase can be dealt with more economically than it appears at the first sight.

The proposed neural network is guaranteed to find the optimal solution of the smooth problem equivalent to the minimization (1). Then, the solution of the original problem can be readily obtained by conducting the subtractions among the outcomes of the neural network. Further, the proposed neural network has a simple one-layer structure that can be smoothly implemented. From the speed point of view, the convergence of the neural network is reliant on a positive parameter determined by the user, not on the size of the dataset. Such a salient feature is desired when large datasets are available. We further adjust the proposed neural network to solve the total variation-regularized problems. Similar to the lasso, the total variation-regularized problems are not differentiable. The efficiency of the proposed neural network is demonstrated by conducting experiments over several real and simulated datasets from the signal and image processing and bioinformatics domain.

In a nutshell, the contributions of this article can be summarized as follows:

- A novel recurrent neural network is proposed for solving the lasso.
- The neural network is guaranteed to converge to the solution of the problem.
- The escalation in dimensions stemming from the variable split is discussed, and the computation cost is reduced.
- The neural network is then extended to solve the total variation-regularized problem.
- Extensive experiments are presented to illustrate the performance of the proposed neural network.

The paper is organized as follows. In Section II, we first derive the smooth problem of the minimization (1), and then a neural network is proposed accordingly. Further, we also analyze the effect of dimension and the complexity of the neural network in this section. The convergence of the neural network and its convergence rate are investigated in Section III. Extensive experimental results with application to compressed sensing and image and signal recovery are discussed in Section IX, and we conclude this paper in Section X.

2. Neural network for smooth equivalent problem

In this section, a smooth problem for the minimization (1) is derived by splitting the desired variable x into its positive and negative parts. The subsequent escalation of dimension and a one-layer neural network are investigated afterward. The proposed neural network is then adjusted to solve the total variation regularized problem.

2.1. Smooth equivalent problem

To solve the minimization (1) using the neural network, we first restate it as a smooth quadratic problem. This is done by splitting variable x into its positive and negative parts. Let $u, v \in \mathbb{R}^n$ be auxiliary variables such that

$$x = u - v \qquad u \geq 0, v \geq 0$$

where $u_i = (x_i)_+, v_i = (-x_i)_+$ and $(.)_+$ denotes the positive part defined as $(x)_+ = max\{0, x\}$. Now, let $\mathbf{1}_{2n} = (1, 1, \ldots, 1) \in \mathbb{R}^{2n}$, then the problem (1) can be rewritten as the following quadratic problem:

$$\begin{aligned} \min_z \quad & F(z) = \frac{1}{2} z^T B z + c^T z \\ s.t. \quad & z \geq 0 \end{aligned} \tag{2}$$

where

$$z = \begin{bmatrix} u \\ v \end{bmatrix}, \quad c = \lambda \mathbf{1}_{2n} + \begin{bmatrix} -A^T y \\ A^T y \end{bmatrix}$$

$$B = \begin{bmatrix} A^TA & -A^TA \\ -A^TA & A^TA \end{bmatrix}$$

2.2. One-layer neural network

The smooth problem (2) is a convex minimization with non-negativity constraints. Therefore, the Karush–Kuhn–Tucker (KKT) conditions [16] are necessary and sufficient for the optimality of the solution. As stated by K.K.T conditions, z^* is the optimal solution of the minimization (2) if and only if there exists $w^* \in R^{2l}$ such that (z^*, w^*) satisfies the following conditions:

$$\begin{cases} \nabla F(z) - w = 0, \ w \geq 0 \\ w^T z = 0, \ z \geq 0. \end{cases} \tag{3}$$

From the first equality in Eq. (3), it is drawn that $\nabla F(z) = w$. The foregoing equations could be thus restated as

$$\nabla F(z) \geq 0, \quad z \geq 0, \quad \nabla F(z)^T z = 0. \tag{4}$$

The inequalities (4) are known as the nonlinear complementarity problem (NCP) [17]. With the aid of the next theorem, a neural network for the minimization (2) is proposed according to the above NCP.

Theorem 2.1. *For the problem (2), z^* is the optimal solution if and only if $\Gamma(z^*) = 0$, where*

$$\Gamma(z) = \min\{z, \nabla F(z)\}, \tag{5}$$

and $\Gamma(z)$ is a vector value function, and "min" represents the minimum value of each element of z and $\nabla F(z)$.

Proof. It can be easily drawn from the inequalities (4) (see [18] for more information). □

Based on the above theorem, the following dynamic system is proposed to solve the problem (2)

$$\frac{dz}{dt} = -\alpha \Gamma(z) \tag{6}$$

where $\alpha > 0$ is a scaling parameter. The dynamic system (6) can be recognized as a recurrent neural network with a single-layer structure. Before examining its structure, however, we first probe into the effect of the dimension escalation caused by the variable split.

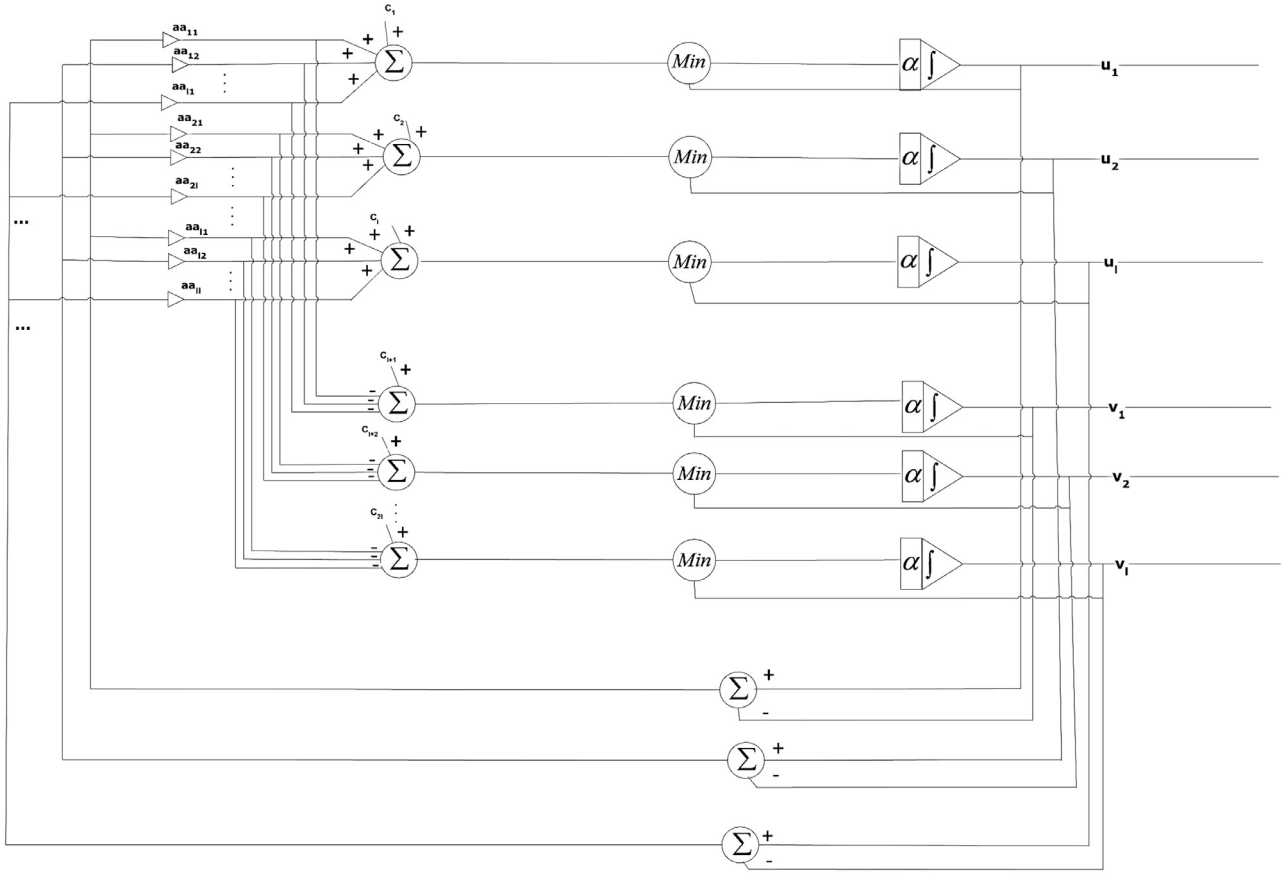


Fig. 1. Block diagram of the proposed recurrent neural network (6) taking the computational reduction into account. The aa_{ij} is the element at the *ith* row and *jth* column of A^TA, and the triangle and Σ represent the multiplication and addition, respectively.

2.3. Dimension effect and complexity of neural network

It is observed that the size of the problem (2) is twice as large as the original problem (1) while $x \in \mathbb{R}^l$ but $z \in \mathbb{R}^{2l}$. However, this increase in dimension does not have a significant impact since the matrix operation to obtain B can be performed more efficiently than it might seem. To illustrate the minority of this effect, let us consider the complexity of the system (6) by computing the number of multiplications and additions/subtractions in each iteration. The most costly computation belongs to Bz while B is a $2l \times 2l$ matrix and $z \in \mathbb{R}^{2l}$. Such a calculation requires $4l^2$ multiplications and $4l^2 - 2l$ additions.

However, the computation can be significantly reduced. For a given $z = (u^T, v^T)^T$, one can rewrite Bz as

$$Bz = B\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} A^TA(u-v) \\ -A^TA(u-v) \end{bmatrix}.$$

The computation of Bz only requires l^2 multiplications and l^2 additions/subtractions, considering that A^TA should be computed beforehand. Hence, the number of operations has dropped from $4l^2$ multiplications to l^2, and from $4l^2 - 2l$ additions/subtractions to l^2. In the aggregate, as c is also a pre-process computation, l^2 multiplications and $l^2 + 2l$ additions/subtractions are done in each iteration of the dynamic system (6).

In the element form, the dynamic system (6) can written as

$$\begin{aligned} \frac{dz_i}{dt} &= \Gamma(z_i) = \min((Bz)_i + c_i, z_i) \\ &= \min(sign(l-i)\sum_j (aa_{ij}(u-v)_j + c_i, z_i) \end{aligned} \tag{7}$$

where aa_{ij} is the element in the *ith* row and *jth* column of the matrix A^TA. As regards the element-wise equation of the proposed neural network, its structure is displayed in Fig. 1. In this figure, the modification for dimension escalation is also considered to reduce the complexity of the network. The outputs of the neural network are u_i's and v_i's, which are recursively entered in the first layer. They are then multiplied by aa_{ij}, which are shown as the triangle in the figure and are explained in Eq. (7). In the view of Fig. 1, the circuit consists of $2l$ integrators, $2l$ activation minimum functions, $4l$ summers, and some connection weights.

2.4. Total variation-regularized problem

The total variation-regularized problem is another non-smooth minimization. The corresponding minimization function for total variation-regularized problem is

$$\min_q \|p-q\|_2^2 + \lambda \|q\|_{TV}$$

where $p \in R^l$ is the observation, $q \in R^l$ is the desired variable, λ is the regularization parameter and $\|x\|_{TV} = \sum_{i=1}^{l-1} |x_i - x_{i+1}|$ is the total variation norm. This problem can be equivalently rewritten as

$$\min_q \|p-q\|_2^2 + \lambda \|Dq\|_1 \tag{8}$$

where $D \in R^{l-1,l}$ is defined as

$$D = \begin{bmatrix} 1 & -1 & 0 & \dots 0 & 0 \\ 0 & 1 & -1 & \dots 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix}.$$

To all appearances, the problem (8) is similar to the minimization (1); however, the total variation-regularized problem has more major challenges as the variable in the l_1-regularization has been multiplied by a matrix.

Harchaoui and Levy-Leduc [19] solved the total variation regularized minimization (8) through the problem (1). The following theorem summarizes their main result.

Theorem 2.2 [19]. *By the following change in variables, the minimizations (1) and (8) are equivalent:*

$$x = Dq$$
$$A = D^T(DD^T)^{-1}$$
$$y = D^T(DD^T)^{-1}Dp \tag{9}$$

where D, p, and q are the variables in the total variation problem. Further, the variable q in the minimization (8) is obtained as

$$q = p + D^T(DD^T)^{-1}(x - Dp). \tag{10}$$

In other words, the total variation-regularized problem (8) can be solved by the minimization (1) with the initialization (9). Then, the optimal solution q is calculated by Eq. (10).

Based on this theorem, the proposed recurrent neural network can be adjusted to solve the total variation-based regularization as well. The major elements for the neural network computation are

$$A^TA = (DD^T)^{-1}$$
$$A^Ty = (DD^T)^{-1}Dp$$

In the experiment section, two applications of the total variation regularization are investigated.

3. Convergence analysis

To assess the reliability of the proposed dynamic system, we first discuss its stability and convergence, and further investigate the properties of the presented RNN. The system is proved to be globally convergent and stable in a Lyapunov sense.

Definition 3.1. A continuous-time neural network is said to be globally convergent if the trajectory of the corresponding dynamic system converges to an equilibrium point for any initial point $z(t_0)$. In other words, the equilibrium z_e is convergent if

$$\exists \delta > 0 \quad s.t. \quad \|z(t_0) - z_e\| < \delta \Longrightarrow \lim_{t \longrightarrow \infty} z(t) = z_e.$$

Lemma 3.2. *The function $\Gamma(.)$, defined in the system (5), is a Lipschitz continuous function. Therefore, there exists a positive constant L such that*

$$\|\Gamma(x) - \Gamma(y)\| \le L\|x - y\|, \quad \forall x, y \in \mathbb{R}^{2n}. \tag{11}$$

Proof. For any arbitrary $x, y \in \mathbb{R}^{2n}$, we have

$$\begin{aligned}
&\|\Gamma(x) - \Gamma(y)\| = \|\min\{x, \nabla F(x)\} - \min\{y, \nabla F(y)\}\| \\
&= \|\frac{x + \nabla F(x) - |x - \nabla F(x)|}{2} - \frac{y + \nabla F(y) - |y - \nabla F(y)|}{2}\| \\
&= \|1/2\{(x - y) + (\nabla F(x) - \nabla F(y)) - |x - \nabla F(x)| \\
&\quad + |y - \nabla F(y)|\}\| \\
&\le 1/2\{\|x - y\| + \|\nabla F(x) - \nabla F(y)\| + \||x - \nabla F(x)| \\
&\quad - |y - \nabla F(y)|\|\} \\
&\le 1/2\{\|x - y\| + \|\nabla F(x) - \nabla F(y)\| + \|x - \nabla F(x) - y + \nabla F(y)\|\} \\
&\le \|x - y\| + \|(Bx + c) - (By + c)\| \\
&= (1 + \|B\|)\|x - y\|,
\end{aligned}$$

Now, let $L = (1 + \|B\|)$ and the proof is complete. □

The upcoming discussion elaborates the convergence and stability of the system (6).

Theorem 3.3. *For any initial point z_0, there exists a unique continuous solution $z(t)$ for (6) within the finite time. Moreover, the equilibrium point of (6) is the solution of the minimization (2).*

Proof. According to Lemma 3.1, the function $\Gamma(z)$ is Lipschitz continuous and so is the right-hand side of the system (6). Thus, based on the Peano's theorem for ODEs [20], there exists a unique continuous solution $z(t)$ for (6) defined on $t_0 \le t \le T_f$. The interval $[t_0, T_f)$ is the so-called maximal interval of existence.

Furthermore, we show that $T_f = \infty$ if the set of all possible solutions, $\Omega = \{z \in \mathbb{R}^{2n} | z \ge 0\}$, is bounded. To do so, let Ω be bounded and $z_0 \in \Omega$; and let $|z - \nabla F(z)|$ represent $(|z_1 - \nabla F(z)_1|, \ldots, |z_{2n} - \nabla F(z)_{2n}|)$. We have

$$\begin{aligned}
\|\Gamma(z)\| &= \|\min\{z, \nabla F(z)\}\| = \|\frac{z + \nabla F(z) - |z - \nabla F(z)|}{2}\| \\
&\le 1/2(\|z + \nabla F(z)\| + \|z - \nabla F(z)\|) \\
&\le 1/2(\|z\| + \|\nabla F(z)\| + \|z\| + \|\nabla F(z)\|) \\
&\le \|z\| + \|\nabla F(z)\|
\end{aligned}$$

On the other hand, since Ω is bounded, there exists a vector K such that for any $z \in \mathbb{R}^n$ we have $\|\nabla F(z)\| \le \|K\|$ ([21]). It is obtainable that

$$\begin{aligned}
\|z(t)\| &\le \|z_0\| + \alpha \int_{t_0}^{t} \|\Gamma(z(s))\| ds \\
&\le \|z_0\| + \alpha \int_{t_0}^{t} \|z(s)\| + \|\nabla F(z)\| ds \\
&\le \|z_0\| + \alpha(\|K\|(t - t_0)) + \int_{t_0}^{t} \|z(s)\| ds)
\end{aligned}$$

Furthermore, by Gronwall inequality [22]

$$\|z(t)\| \le \|z_0\| + \alpha\|K\|(t - t_0)\exp(\alpha(t - t_0)).$$

Thus, the solution $z(t)$ is bounded on $[t_0, T_f)$, which implies $T_f = \infty$ and this completes the proof of the first part.

Now, if z^* is the equilibrium point of system (6), then $\Gamma(z^*) = 0$, and according to Theorem 2.1 his equilibrium point is the optimal solution of problem (2). □

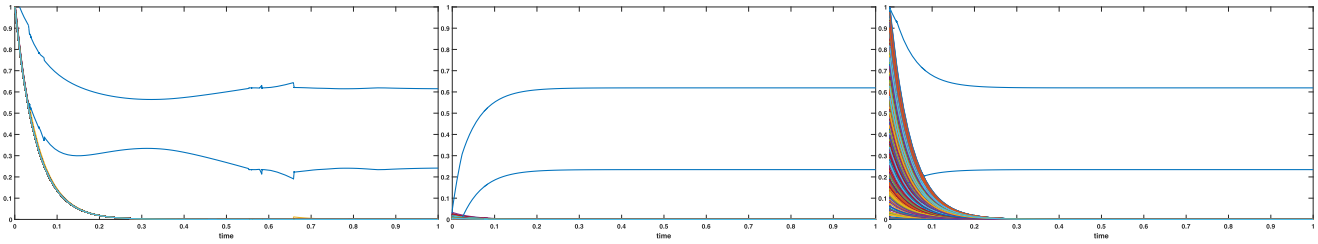
Theorem 3.4. *The proposed neural network (6) with the initial point $z_0 \in \mathbb{R}^{2n}$ is stable in the sense of Lyapunov and globally converges to the solution of (2). Moreover, the convergence rate of the neural network (6) escalates as α increases.*

Proof. According to Theorem 3.1, there exists a unique solution z^* for the system (6) within the interval $[t_0, T_f)$. Let $z \in \Omega$ and consider the following Lyapunov function:

$$E(z) = F(z) - F(z^*).$$

It is readily seen that $E(z) \ge 0$ because z^* is the optimal solution of the minimization (2). Further, z^* is the optimal solution of problem (2) if and only if $\Gamma(z^*) = 0$ (according to Theorem 2.1), and the solution of $\Gamma(z) = 0$ is unique (by Theorem 3.3), so is the solution of the problem (2). Thus, $E(z) = 0$ if and only if $z = z^*$. Moreover, we have

$$\begin{aligned}
\frac{dE(z)}{dt} &= \left(\frac{dE(z)}{dz}\right)^T \frac{dz}{dt} \\
&= -\alpha\nabla F(z)^T(\Gamma(z)) \\
&= -\alpha\nabla F(z)^T(\frac{z + \nabla F(z) - |z - \nabla F(z)|}{2}) \\
&= -\frac{\alpha}{2}(\nabla F(z)^T z + \|\nabla F(z)\|^2 - \nabla F(z)^T|z - \nabla F(z)|) \\
&\le \frac{\alpha}{2}(-\nabla F(z)^T z - \|\nabla F(z)\|^2 + \nabla F(z)^T|z| + \|\nabla F(z)\|^2) = 0,
\end{aligned} \tag{12}$$



(a) With the initialization z=1 (b) With the initialization z=0 (c) With the random initialization

Fig. 2. Convergence of the proposed neural network (6) with $\alpha = 10$ and different initializations: (a) with the initialization $z = 1$;(b) with the initialization $z = 0$; (c) with the random initialization. The x-axis is the iteration and y-axis is the value of elements of the desired variable x in the lasso problem.

where $|z| = z$ since $z \geq 0$. Hence, the system (6) is stable in the sense of Lyapunov. We further investigate the global convergence of the proposed system and show that $dz/dt = 0$ if and only if $dE/dt = 0$. To do so, let $dz/dt = 0$ which implies $\Gamma(z) = 0$, then clearly

$$\frac{dE}{dt} = -\alpha \nabla F(z)^T \Gamma(z) = 0.$$

Conversely, if $dE/dt = 0$, then

$$\nabla F(z)^T (\Gamma(z)) = 0.$$

In this equation, $\Gamma(z) = 0$ results in $dz/dt = 0$ and the proof is complete. But if $\Gamma(z) \neq 0$ and $\nabla F(z) = 0$, we get (since $z \geq 0$)

$$\frac{dz}{dt} = -\alpha \Gamma(z) = \min\{z, \nabla F(z)\} = \nabla F(z) = 0.$$

Therefore, the presented system (6) is stable in the sense of Lyapunov and globally converges to the optimal solution of (2).

Moreover, the inequality in (12) implies that as α increases, the convergence rate also increases. □

4. Experiment results

This section presents the experimental results regarding the proposed neural network. First, the convergence analysis of the neural network was empirically investigated, and its dependency on the parameter α was verified. Then, the proposed neural network was applied to three different applications. The first was to recover a sparse signal from noisy observations. The other two were an image restoration and an aCGH data recovery, in which the total variation-regularized minimization is utilized. The proposed neural network is implemented in MATLAB by the ordinary differential equations (ODE) solvers.

4.1. Empirical convergence analysis

The convergence of the proposed neural network has been theoretically investigated. We now present empirically exploration of the convergence of the proposed neural network (6) as a complement to the theoretical studies. To do so, the *WINE* benchmark problem, which consists of 178 data with four attributes, was selected. To check the convergence, y was set to one of the data points randomly selected from the dataset, and A was the remaining data. Thus, the minimization of the problem (1) obtained a coefficient vector that enabled us to write the randomly selected sample as a linear combination of other data points. This is known as the *self-expressiveness property*, which is utilized in recent works [5,23]. The convergence is scrutinized by various initializations in order to check the sensitivity of the neural network to the initialization. Let $\alpha = 10$, Fig. 2 plots the convergence of the neural network trajectory with the initial point $z = [1, \ldots, 1] \in \mathbb{R}^{356}$, $z = [0, \ldots, 0] \in \mathbb{R}^{356}$ and random initialization, respectively. The x-axis in this figure is the iteration and y-axis is the value of each element of the vector x. In this figure, it is clear that most of the coefficients z converge to zero, which is the reason for the l_1-regularization. Further, the non-zero coefficients converge to the same values (one around 0.22 and another around 0.61). This indicates that the neural network is globally convergent to the optimal solution, and its convergence is not reliant on the initializations.

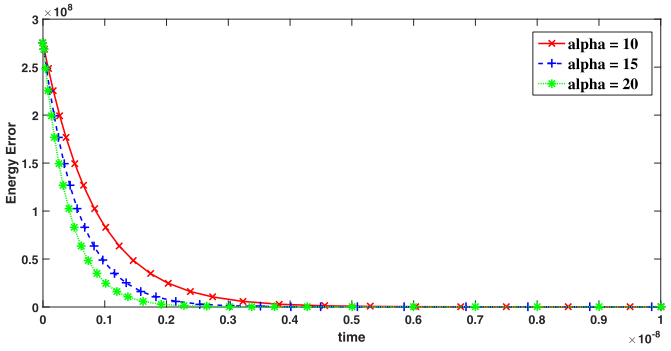


Fig. 3. The transient behavior of the energy error based on the neural network (6) for three different values of α on the *WINE* benchmark. The solid, dashed and dotted lines correspond to $\alpha = 10$, 15 and 20, respectively.

Furthermore, we explored the convergence rate behavior of the neural network (6). To do so, we repeated the previous experiment over the *WINE* benchmark and assumed that α is 10, 15 and 20 in the dynamic system (6). The energy error of the proposed neural network can be defined as

$$ER(z) = \|\Gamma(z)\|_2. \tag{13}$$

According to the dynamic system (6), $ER(k^*) = 0$ if and only if k^* is an optimal solution. Fig. 3 shows the transient behavior of the error. It is readily observable that the bigger value of α accelerates the convergence of the proposed neural network on the same problem. Thus, one can accelerate the convergence simply by increasing the parameter α.

4.2. Signal reconstruction

In this section, we consider a sparse signal recovery problem with a signal $x \in \mathbb{R}^{4096}$. In this example (shown at the top of Fig. 4), there are 160 spikes with ± 1 amplitude. The matrix $A \in \mathbb{R}^{1024 \times 4096}$ is filled with independent samples of the standard normal distribution with orthonormalized rows. The observation y is generated according to

$$y = Ax + n \tag{14}$$

where n is a noise drawn according to the normal distribution $N(0, 0.01)$ on $\mathbb{R}^{1024}$. The parameter λ is also chosen by

$$\lambda = 0.01 \|A^T y\|_\infty; \tag{15}$$

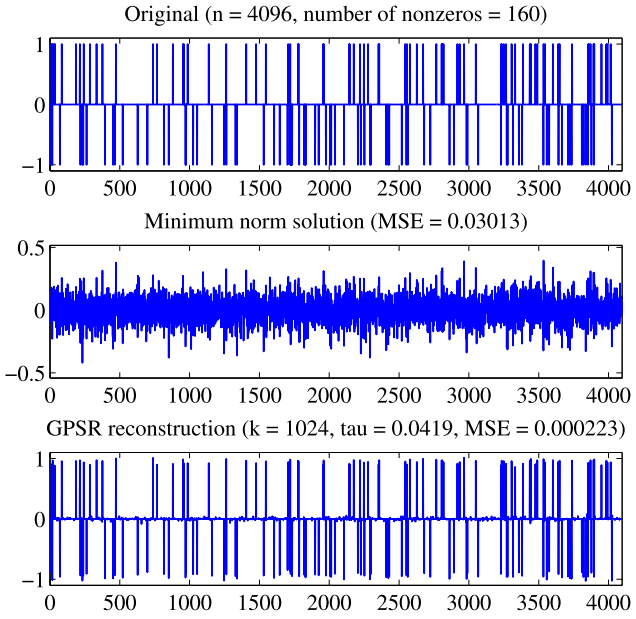


Fig. 4. Sparse signal reconstruction. Top: the original signal. Middle: the minimum energy reconstruction. Bottom: the reconstructed signal using the neural network (6).

as for $\lambda > \|A^T y\|_\infty$, the unique minimum of (1) is the zero vector [24].

Fig. (4) shows the reconstruction results. The original signal is presented at the top of the plot. The middle plot shows the signal $x = A^\dagger y$, which is known as the minimum energy reconstruction. The bottom plot delineates the reconstructed signal by the proposed neural network (6). As can be readily grasped form this figure, the proposed neural network can faithfully recover the corrupted signal even though only a few of the non-zero measurements are available in comparison to all elements.

4.3. aCGH data recovery

Array comparative genome hybridization (CGH array or aCGH) is a new technique to discover the aberration in the DNA copy number [25,26]. The greatest challenge in finding the aberrations is that aCGH data are highly corrupted by various noises so that the boundaries of the normal and aberrant genomes cannot be readily detected. As a result, it is of the utmost importance to remove the noises from the raw aCGH data prior to the aberration detection.

The most popular way of denosing aCGH data is to solve a problem regularized by the total variation norm. These methodologies process either all the aCGH samples in a dataset simultaneously [27–30] or each sample separately [31,32].

We applied the proposed neural network for noise removal from the aCGH data and compared it with state-of-the-art algorithms such as total variation and spectral regularization (TVSp) [33], piece-wise and low rank approximation (PLA) [34], low rank recovery based on the half-quadratic minimization (LRHQ) [30], and group fused lasso segmentation (GFLseg) [28]. TVSp takes advantage of the nuclear norm regularization along with the total variation norm. By the same token, PLA and LRHQ have similar formulation, with more sparsity constraints in the former method and more robust information-theoretic loss function in the latter method. GFLseg is yet another technique that utilizes the weighted $l1 - l2$ norm with the integral total-variation regularization. All of these methods have more parameters to be tuned (at least two), and are of higher complexity due to the various regularizations they employed. In the following, we show that the proposed neural network is competitive with the state of the art despite its simplicity and lower number of parameters.

The performance comparison was twofold. First, The comparison was conducted based on receiver operating characteristic (ROC) curves across simulated datasets contaminated by different types of noise. Second, two real-world aCGH datasets were used to carry out the recovery.

4.3.1. Experiment on simulated data

In this subsection, the methods mentioned above are compared across synthesized datasets. In the experiment, 50 samples with a length of 500 were generated according to the methodology presented in [33]. The simulated data were corrupted by a Gaussian noise with different signal-to-noise (SNR) ratios. For the first comparison, we plot the ROC diagram for the methods. The ROC is a curve plotting the true positive rate (TPR) against the false positive rate (FPR) for different thresholds. Given a threshold T, the true and false positive rates are defined as

$$TPR(T) = \frac{|P_T|}{|A|} \quad FPR(T) = \frac{|FP_T|}{|N|}$$

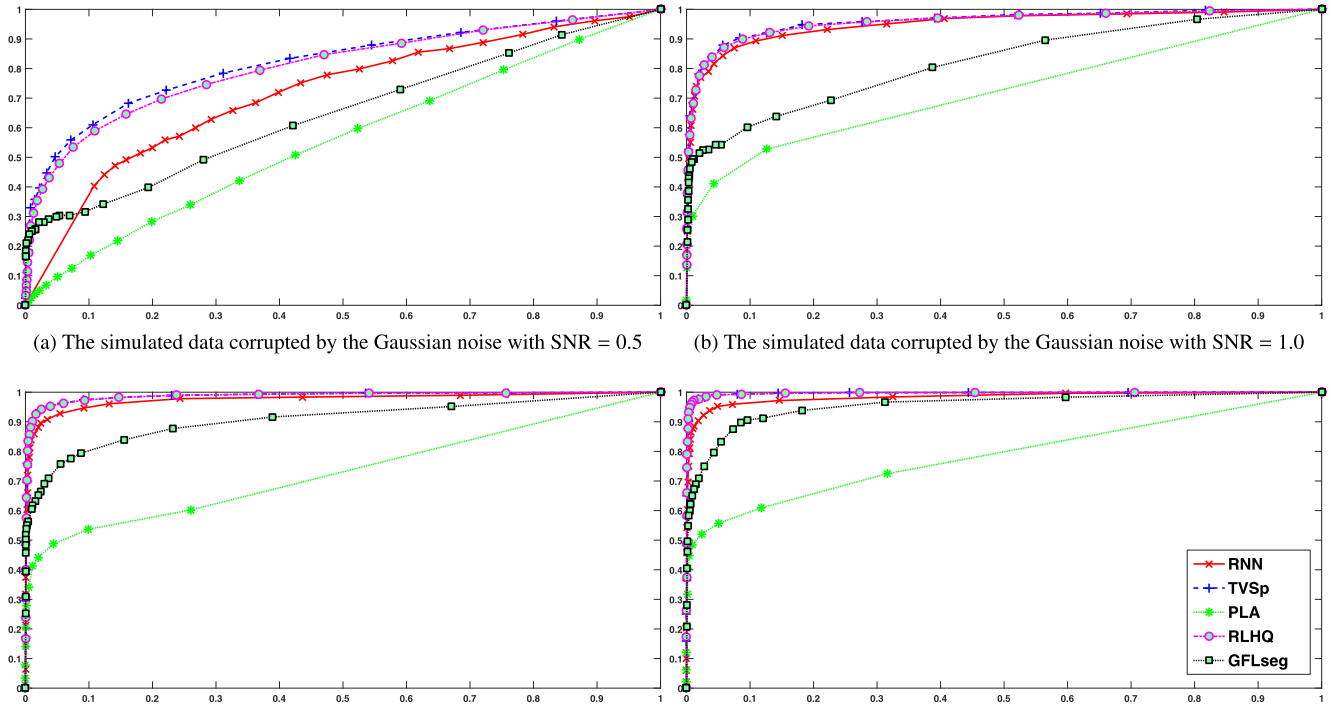
where A and N are respectively real aberrations and normal genomes, P_T and FP_T are respectively the truly and falsely discovered aberrations, and |.| is the cardinality operator. These elements can be easily obtained as the study was on the simulated data. In the ROC curve, more deviation from the diagonal indicates the superiority of the methods. Fig. 5 plots the ROC diagram for different SNRs. The proposed neural network consistently outperforms PLA and GFLseg in all scenarios as it has more digression from the diagonal. However, TVSp and LRHQ are slightly better than the proposed neural network. For $SNR = 0.5$, the superiority of TVSp and LRHQ is more evident while the proposed neural network is competitive for other SNRs. The reason for such a difference is the complexity of TVSp and LRHQ. Both utilize the nuclear norm (besides the total variation) in their problem to induce the low rank in the recovered profiles. Such a regularization increases the complexity and requires the interminable singular value decomposition in each iteration. Despite its simplicity, the recurrent neural network has a reasonable performance in removing the noise from aCGH data.

4.3.2. Experiment on real datasets

The performance of the proposed neural network was then investigated across real datasets. To do so, two datasets were employed: the Pollack et al. dataset [35], which includes 44 breast tumors of 6691 human mapped genes, and Chin et al. dataset [36], which consists of 2149 clones from 141 primary breast tumors.

These datasets were subjected to the proposed neural network to obtain the recovered profiles. Fig. 6 plots the heat and bar diagrams for the retrieved profiles of the datasets mentioned above. The heat maps are plotted at the top and the bar diagram, which is the sum of the number of grains across all samples given a threshold, is at the bottom. As the color bar suggests, the yellowish segments in the heat map indicate the duplication and the bluish segments indicate the loss in the aCGH data. The greenish parts, which are indeed prevalent in the heat map, are where there is no aberration. The results from the bar diagrams indicate that probes 178–184 from the Pollack et al. dataset and probes 38–39 from the Chin et al. dataset are amplification regions. Regarding their locations on the chromosome, the discovered areas from both datasets are in accordance with each other and are also in line with other studies on breast cancer [35,36].

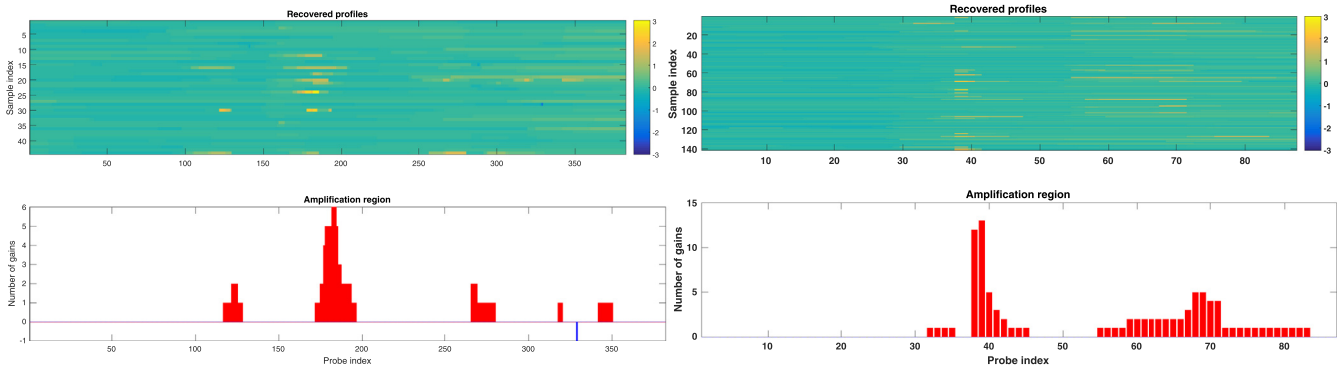
To show the efficient data recovery by the neural network, several recovered profiles from the proposed neural network, TVSp [33] and PLA [34] are presented in Fig. 7. Each column



(a) The simulated data corrupted by the Gaussian noise with SNR = 0.5 (b) The simulated data corrupted by the Gaussian noise with SNR = 1.0

(c) The simulated data corrupted by the Gaussian noise with SNR = 1.5 (d) The simulated data corrupted by the Gaussian noise with SNR = 2.0

Fig. 5. The performance comparison of the proposed recurrent neural network (RNN), TVSp [33], PLA [34], LRHQ [30] and GFLseg [28] via the ROC curve. The ROC curves of different methods on the simulated data corrupted by the Gaussian noise with different SNRs: (a) SNR = 0.5; (b) SNR = 1.0; (c) SNR = 1.5; (d) SNR = 2.0. The *x*-axis and *y*-axis of each figure is the false positive rate and the true positive rate, respectively.



(a) The Pollack et al. dataset [35] (b) The Chin et al. data [36]

Fig. 6. The profiles retrieved by the proposed neural network; (a) the recovered profiles of the Pollack et al. dataset [35]; (b) the recovered profiles of the Chin et al. dataset [36]. The yellowish color in the heat map (the top figure) indicates the duplication and the bluish shows the loss in the chromosome. The greenish areas are the normal regions. The bottom is the bar diagram which plots the sum of the number of aberrations with the threshold 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in this figure is dedicated to one sample, and each column corresponds to a recovery method. Further, the red dots are the real data, and the blue lines indicate the data recovered by each method. From the smoothness perspective, the proposed neural network consistently outperforms PLA and TVSp, since the recovered data are much smoother than those recovered by PLA and TVSp.

4.3.3. Time complexity

The proposed neural network was empirically evaluated in terms of the execution time. To this end, 50 aCGH samples with a different number of probes were generated and corrupted with a random Gaussian noise. The resulting corrupted data were then subjected to different methods for recovery, and the time needed to do so is the parameter based on which the various algorithms are contrasted. The numbers of probes for this experiment were 50, 500, 1000, and 10,000. The experiments were performed on a PC with a 3.2 Core-i5 CPU and 4 GB of RAM.

Fig. 8 plots the time in seconds that each method needed to complete the recovery task with different numbers of probes. The proposed neural network significantly outperforms RCLR, and is quite competitive with TVSp. PLA and GFLSeg are much faster than the others, mainly due to the fact they have implemented a part of their algorithm in C/C++, which is inherently swift.

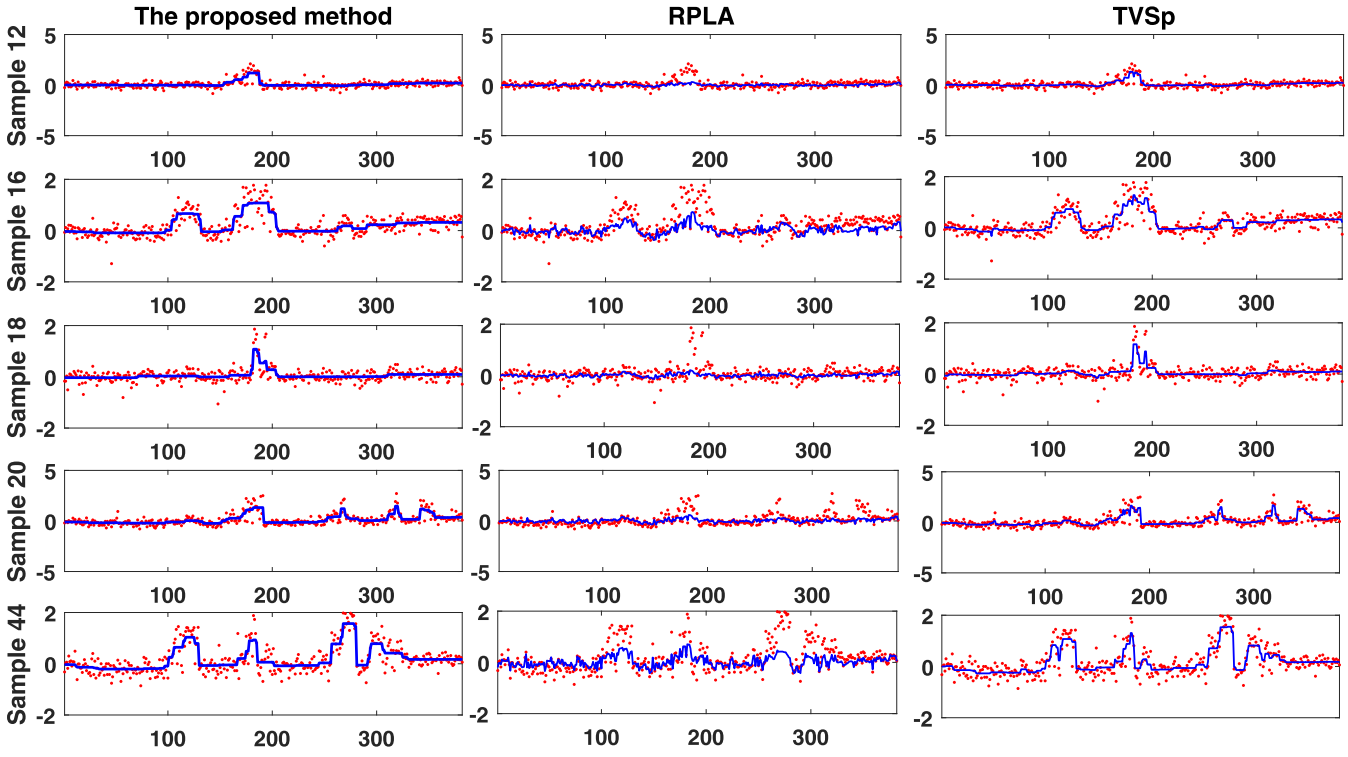


Fig. 7. Five selected samples from the Pollack et al. dataset recovered by various methods. Each row in this figure corresponds to a sample and each column tallies with a recovery method. The three methods are the proposed neural network, PLA [34] and TVSp [33]. The red dots are the real data from the datasets, and the blue lines are the data retrieved by each method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

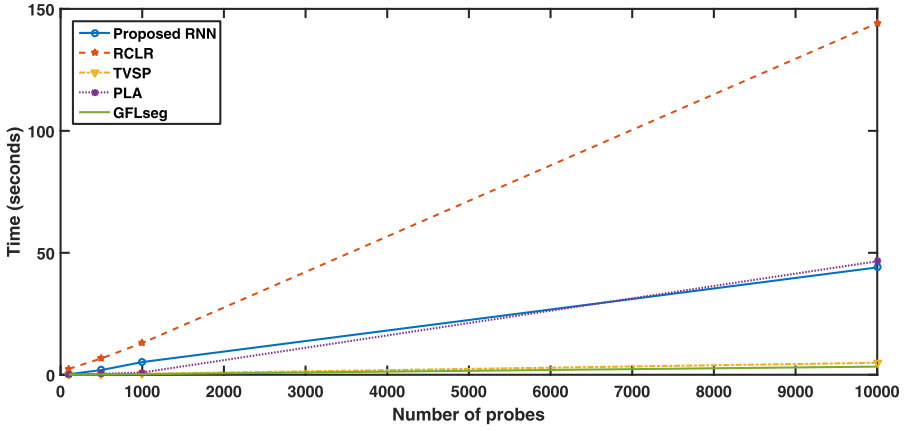


Fig. 8. The time required for each method to complete the recovery task over a dataset with 50 sample and different numbers of probes. The x-axis is the number of probes, and y-axis is the time in seconds for each method to complete the task.

4.4. Image restoration

The final experiment was to recover the original image from noisy observations. To do so, three images were selected and contaminated by the Gaussian noise with $\sigma = 0.05$. The first and second columns of Fig. 9 correspond to the original and noisy images under study, respectively. The total variation-regularized problem (8) was utilized to recover the original images from the contaminated observations. The recovery was carried out by the proposed neural network and the primal-dual splitting method (PDSM) [37]. The images recovered by PDSM and the proposed neural network are presented in the third and fourth columns, respectively. This figure clearly shows that the proposed neural network has faithfully recovered the images. We further tabulate the mean square error of two methods for each image in Table 1. The table also confirms that the proposed neural network retrieves the original images with high confidence and is competitive with PDSM.

Table 1
The mean square errors of the proposed neural network and the primal-dual splitting method (PDSM) [37] across three images.

Image	RNN	PDSM
MRI	3.08×10^{-3}	5.75×10^{-5}
Lena	6.49×10^{-5}	9.13×10^{-5}
Cameraman	7.47×10^{-5}	9.54×10^{-5}

Fig. 9. Image recovery by the proposed neural network and PDSM [37]. The columns from left to right correspond to the original image, noisy image, the image recovered by PDSM, and the image retrieved by the neural network, respectively.

5. Conclusion

This paper presented a one-layer recurrent neural network to find the optimal solution of the l_1-regularized least square problem. The proposed neural network is guaranteed to globally converge to the solution of this problem while its convergence is reliant not upon the size of the datasets but upon a constant parameter. The experiments further investigated the convergence of the neural network and its dependence on the constant parameter. The proposed recurrent neural network was applied to several problems including sparse signal recovery, image restoration, and aCGH data recovery. These applications showed the reasonable performance of the proposed neural network in comparison with other state-of-the-art methods.

References

[1] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Royal Stat. Soc. Ser. B (Methodol.) 58 (1) (1996) 267–288.
[2] S.J. Wright, R.D. Nowak, M.A. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process. 57 (7) (2009) 2479–2493.
[3] C.M. Bishop, et al., Pattern Recognition and Machine Learning, 1, Springer, New York, 2006.
[4] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, IEEE, 2009, pp. 2790–2797.
[5] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[6] H. Lee, A. Battle, R. Raina, A.Y. Ng, Efficient sparse coding algorithms, in: Proceedings of Advances in Neural Information Processing Systems, 2006, pp. 801–808.
[7] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 689–696.
[8] L. Jin, S. Li, X. Luo, Y. Li, B. Qin, Neural dynamics for cooperative control of redundant robot manipulators, IEEE Trans. Neural Netw. Learn. Syst. (2018).
[9] M.A. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 586–597.
[10] J. Kim, H. Park, Fast active-set-type algorithms for l1-regularized linear regression, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2010, pp. 397–404.
[11] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale l 1-regularized least squares, IEEE J. Sel. Top. Signal Process 1 (4) (2007) 606–617.
[12] Y. Xiao, Q. Wang, Q. Hu, Non-smooth equations based method for 1-norm problems with applications to compressed sensing, Nonlinear Anal.: Theory Methods Appl. 74 (11) (2011) 3570–3577.
[13] P.G.C. Zhang, in: A Fast Dual Projected Newton Method for L1-Regularized Least Squares, Tsinghua University, Beijing, 2011.
[14] P. Tseng, S. Yun, A coordinate gradient descent method for nonsmooth separable minimization, Math. Program. 117 (1–2) (2009) 387–423.
[15] I. Loris, M. Bertero, C. De Mol, R. Zanella, L. Zanni, Accelerating gradient projection methods for 1-constrained signal recovery by steplength selection rules, Appl. Comput. Harmon. Anal. 27 (2) (2009) 247–254.
[16] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear programming: Theory and Algorithms, John Wiley & Sons, 2013.
[17] O.L. Mangasarian, Equivalence of the complementarity problem to a system of nonlinear equations, SIAM J. Appl. Math. 31 (1) (1976) 89–92.
[18] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, 23, Prentice Hall, Englewood Cliffs, NJ, 1989.
[19] C. Levy-leduc, Z. Harchaoui, Catching change-points with lasso, in: Proceedings of Advances in Neural Information Processing Systems, 2008, pp. 617–624.
[20] J.K. Hale, Functional Differential Equations, Springer, 1971.
[21] S. Boyd, A. Mutapcic, Subgradient Methods, in: Notes for EE364b, Stanford University, Winter 2006-07.
[22] R. Bellman, et al., The stability of solutions of linear differential equations, Duke Math. J. 10 (4) (1943) 643–647.
[23] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 663–670.
[24] J.-J. Fuchs, On sparse representations in arbitrary redundant bases, IEEE Trans. Inf. Theory, 50 (6) (2004) 1341–1344.

[25] D. Pinkel, D.G. Albertson, Array comparative genomic hybridization and its applications in cancer, Nat. Genet. 37 (2005) S11–S17.
[26] L. Feuk, A.R. Carson, S.W. Scherer, Structural variation in the human genome, Nat. Rev. Genet. 7 (2) (2006) 85–97.
[27] C.M. Alaíz, Á. Barbero, J.R. Dorronsoro, Group fused lasso, in: International Conference on Artificial Neural Networks, Springer, 2013, pp. 66–73.
[28] K. Bleakley, J.-P. Vert, The group fused lasso for multiple change-point detection, arXiv preprint arXiv:1106.4199 (2011).
[29] H.S. Noghabi, M. Mohammadi, Y.-H. Tan, Robust group fused lasso for multisample copy number variation detection under uncertainty, IET Syst. Biol. 10 (6) (2016) 229–236.
[30] M. Mohammadi, G.A. Hodtani, M. Yassi, A robust correntropy-based method for analyzing multisample aCGH data, Genomics 106 (5) (2015) 257–264.
[31] A. Mitra, G. Liu, J. Song, A genome-wide analysis of array-based comparative genomic hybridization (CGH) data to detect intra-species variations and evolutionary relationships, PloS one 4 (11) (2009) e7978.
[32] J. Hu, J.-B. Gao, Y. Cao, E. Bottinger, W. Zhang, Exploiting noise in array CGH data to improve detection of DNA copy number change, Nucl. Acids Res. 35 (5) (2007) e35.
[33] X. Zhou, C. Yang, X. Wan, H. Zhao, W. Yu, Multisample ACGH data analysis via total variation and spectral regularization, IEEE/ACM Trans. Comput. Biol. Bioinform. 10 (1) (2013) 230–235.
[34] X. Zhou, J. Liu, X. Wan, W. Yu, Piecewise-constant and low-rank approximation for identification of recurrent copy number variations, Bioinformatics 30 (14) (2014) btu131.
[35] J.R. Pollack, T. Sørlie, C.M. Perou, C.A. Rees, S.S. Jeffrey, P.E. Lonning, R. Tibshirani, D. Botstein, A.-L. Børresen-Dale, P.O. Brown, Microarray analysis reveals a major direct role of dna copy number alteration in the transcriptional program of human breast tumors, Proc. Natl. Acad. Sci. 99 (20) (2002) 12963–12968.
[36] K. Chin, S. DeVries, J. Fridlyand, P.T. Spellman, R. Roydasgupta, W.-L. Kuo, A. Lapuk, R.M. Neve, Z. Qian, T. Ryder, et al., Genomic and transcriptional aberrations linked to breast cancer pathophysiologies, Cancer Cell 10 (6) (2006) 529–541.
[37] L. Condat, A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms, J. Optim. Theory Appl. 158 (2) (2013) 460–479.

Majid Mohammadi is a Ph.D. candidate at Information and Communication Technology group of the Department of Technology, Policy and Management of the Delft University of Technology. He has obtained his B.Sc. and M.Sc. in Software Engineering and Artificial Intelligence, respectively. His main research interest is semantic interoperability, machine learning and pattern recognition.

Yao-Hua Tan is professor of Information and Communication Technology at the ICT Group of the Department of Technology, Policy and Management of the Delft University of Technology and part-time professor of Electronic Business at the Department of Economics and Business Administration of the Vrije university Amsterdam. His research interests are service engineering and governance; ICT-enabled electronic negotiation and contracting; multi-agent modelling to develop automation of business procedures in international trade.

Wout Hofman is senior research scientist at TNO, the Dutch organization for applied science, on the subject of interoperability with a specialization in government (e.g. customs) and business interoperability in logistics. He is responsible for coordinating semantic developments within the iCargo project. Wout is also as member of the Scientific Board of the EU FP7 SEC Cassandra project responsible for IT developments in that latter project.

S. Hamid Mousavi was born in Mashhad, Iran on February 3, 1988. He received the B.Sc. degree in pure mathematics from Ferdowsi University of Mashhad (FUM) in 2011. He started his M.Sc. in applied mathematics in FUM and worked on control and optimization problems. After graduation in 2015, he joined the machine learning group at the University of Oldenburg, Germany, where he is currently working toward a doctorate degree. His major fields of interest currently are optimization and probabilistic algorithms.