EURASIP Journal on Image and Video Processing

RESEARCH

Open Access

The optimally designed autoencoder network for compressed sensing

Zufan Zhang[1], Yunfeng Wu[1], Chenquan Gan[1*] and Qingyi Zhu[2]

Abstract

Compressed sensing (CS) is a signal processing framework, which reconstructs a signal from a small set of random measurements obtained by measurement matrices. Due to the strong randomness of measurement matrices, the reconstruction performance is unstable. Additionally, current reconstruction algorithms are relatively independent of the compressed sampling process and have high time complexity. To this end, a deep learning based stacked sparse denoising autoencoder compressed sensing (SSDAE_CS) model, which mainly consists of an encoder sub-network and a decoder sub-network, is proposed and analyzed in this paper. Instead of traditional linear measurements, a multiple nonlinear measurements encoder sub-network is trained to obtain measurements. Meanwhile, a trained decoder sub-network solves the CS recovery problem by learning the structure features within the training data. Specifically, the two sub-networks are integrated into SSDAE_CS model through end-to-end training for strengthening the connection between the two processes, and their parameters are jointly trained to improve the overall performance of CS. Finally, experimental results demonstrate that the proposed method significantly outperforms state-of-the-art methods in terms of reconstruction performance, time cost, and denoising ability. Most importantly, the proposed model shows excellent reconstruction performance in the case of a few measurements.

Keywords: Compressed sensing, Stacked sparse denoising autoencoder, Deep learning, Multiple nonlinear measurement, Signal reconstruction

1 Introduction

With the increasing demand of information processing, the information sampling rate and device processing speed of signal processing framework are also getting higher increasingly. To reduce the cost of storage, processing, and transmission of massive information, Donoho [1] first proposed a compressed sensing (CS) method, which merges sampling and compression steps. The traditional method samples the data uniformly and then compresses them. The CS method just needs to store and transmit a few non-zero coefficients, which further reduces the time of data acquisition and the complexity of sampling process.

CS method has been applied successfully in many fields, such as biomedical [2, 3], image processing [4, 5], communication [6, 7] and sensor network [8, 9], but there are still two significant challenges in the CS that need to be resolved. On the one hand, due to the strong randomness of measurement matrix, it is difficult to realize measurement matrix on the hardware and the reconstruction performance is unstable. On the other hand, current high-performance reconstruction algorithms just take into account the recovery process while the connection with compressed sampling process is neglected.

For the former challenge, it is critical to design a suitable measurement matrix in the process of compressed sampling. Generally, measurement matrices are divided into random matrices and definite matrices. On the one hand, Gaussian [10] and Bernoulli [11] random matrices are used as the sampling matrices in most previous works because they meet the restricted isometry property [12] with a large probability. However, it always suffers some problems such as high computation cost, vast storage, and uncertain reconstruction qualities. On the other hand, definite matrix had been proposed as an alternative solution to reduce the high cost problem of random matrix, such as Toeplitz polynomial [13] and Circulant matrice [14]. However, the reconstruction quality of the

*Correspondence: gcq2010cqu@163.com
[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
Full list of author information is available at the end of the article

definite matrix is worse than a random matrix. A gradient descent method [15] is designed to minimize the mutual coherence of measurement matrix, which is described as absolute off-diagonal elements of the corresponding Gram matrix. Gao et al. [16] designed a local structural sampling matrix for block-based CS coding of natural images by utilizing the local smooth property of images. As previously mentioned, these measurement matrices still have some disadvantages because they are not optimally designed for signals, and neglect the structure of the signals.

For the latter challenge, the most crucial work in CS is to construct a stable reconstruction algorithm with low computational complexity and less restriction on the number of measurements to accurately recover signals from measurements. According to the volume of data, the CS reconstruction algorithms are divided into two categories: hand-designed recovery methods and data-driven recovery methods. Most of the existing algorithms can be considered "hand-designed" in the sense that they use some sort of expert knowledge, i.e., prior, about the structure of $\mathbf{x}$. The hand-designed methods have three directions: convex optimization, greedy iterative, and Bayesian. Convex optimization algorithms get the approximate solution by translating the non-convex problem into a convex one, e.g., basis pursuit denoising (BPDN) algorithm [17], the minimization of total variation (TV) [18]. Greedy iterative algorithms approach gradually the original signal by selecting a local optimal solution in iteration, e.g., orthogonal matching pursuit (OMP) [19], compressive sampling matching pursuit (CoSaMP) [20], and iterative hard thresholding (IHT) [21]. Bayesian algorithms solve the sparse recovery problem by taking into account a prior knowledge of the sparse signal distribution, e.g., Bayesian via Laplace Prior (BCS-LP) [22] and Bayesian framework via Belief Propagation (BCS-BP) [23]. Unfortunately, these algorithms are too slow for many real-time applications and the potential information in training data is typically underutilized [24]. The second category is data-driven method that builds deep learning framework to solve the CS recovery problem via learning the structure within the training data. For instance, the SDA model [25] was proposed to recover structured signal by capturing statistical dependencies between the different elements of certain signals. Another reference [26] used RBM-OMP-like and DBN-OMP-like CS model, which are based on restricted Boltzmann machines and deep belief network, respectively, to model the prior distribution of the sparsity pattern of signals. Other work in this area used either DeepInverse [27] based on convolutional neural network or ReconNet [28] based on combination of convolutional and fully connected layers to solve the CS recovery problem. The data-driven methods can compete with state-of-the-art methods in terms of performance while running hundreds of times faster compared to the hand-designed methods. However, they need a lot of time and data to train model. The main reason is that these previous methods only consider the recovery process, ignoring the connection with compressed sampling process.

Noting the above discussions and previous work, this paper proposes a SSDAE_CS model based on sparse autoencoder (SAE) [29, 30] and denoising autoencoder (DAE) [31, 32] to solve the two important issues in CS. This model mainly consists of an encoder sub-network and a decoder sub-network. Given enough training data, neural network is acted as universal function approximator to represent arbitrary functions. Thus, two sub-networks are used to learn the mapping functions of the compressed sampling and the recovery process, respectively. A trained encoder sub-network, which uses multiple nonlinear measurements and specially designs for the type of signals, is used to obtain measurements during the compressed sampling process (addressing problem one). Then, these traditional signal reconstruction algorithms are replaced with a trained decoder sub-network to recover original signals from measurements. It just needs a few times matrix-vector multiplications and nonlinear mapping; hence, the proposed approach reduces the time cost in the reconstruction process. In the previous CS researches, the compression process was relatively independent of the recovery process. For this motivation, SSDAE_CS method integrates compressed sampling and recovery processes into a deep learning network to strengthen the connection. Through end-to-end training, the two sub-networks can be jointly optimized to improve the overall performance of CS, but they can also extend to different scenarios as two independent networks (addressing problem two). Finally, experiment results demonstrate that the proposed model significantly outperforms state-of-the-art methods in terms of reconstruction performance, time cost, and denoising ability. Especially, the SSDAE_CS model shows the excellent performance of signal reconstruction in the case of a few measurements.

The rest of this paper is organized as follows: a deep learning model for compressed sensing and the training method of model is presented in Section 2. Experiment results for the proposed method and comparisons with other CS reconstruction algorithms are performed in Section 3. Finally, Section 4 includes the conclusion of this work.

2 Methods

In this section, a deep learning CS model, which integrates the advantages of denoising and sparse autoencoders into CS theory, will be introduced in detail. The following notations are used throughout this paper: boldfaced capital letters such as $\mathbf{W}$ for matrices, small letters such as $\mathbf{x}$ are reserved for vectors and scalars, and italic small letters such as x_i denotes the i^{th} element of the vector. $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ denote the weight matrix and the bias vector associated between layer l and layer $l+1$, respectively. $\mathbf{a}^{(l)}$ denotes the feature vector of the l hidden layer. $f(\cdot)$ represents the activation function, and the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ is used as the activation function.

2.1 Overall framework of SSDAE_CS model

This paper proposes a deep learning model named stacked sparse denoising autoencoder compressed sensing which integrates advantage of denoising and sparse autoencoders into CS theory. The corrupted input is trained to reconstruct the clean version through DAE. In SAE, sparse regularization inhibits the activity of neurons to improve the overall performance of the model. It is similar to the human brain that a small number of neurons are activated and most neurons are inhibited.

As discussed above, traditional CS methods consist of two steps including linear measurement sampling and non-linear reconstruction algorithm. As shown in Fig. 1, our proposed model contains two corresponding modules: an encoder sub-network and a decoder sub-network. Instead of traditional linear measurements, a multiple nonlinear measurement encoder sub-network is trained to obtain measurements during the compressed sampling process. Then, these traditional signal reconstruction algorithms are replaced with a trained decoder sub-network to recover original signals from measurements. The framework of the proposed SSDAE_CS model constituting the training stage and testing stage is illustrated in the upper part of Fig. 1. The training stage is employed to learn a prior parameter for encoder and decoder sub-networks. When trained on a set of representative signal, the network learns both a feature representation to obtain measurements and an inverse mapping to recover signals. The goal of training stage is to learn the optimal encoder and the signals recovery decoder simultaneously. At the test stage, the test set is fed into the training model to test the performance of the model in all aspects.

2.2 Encoder and decoder sub-networks

The architecture of the SSDAE_CS model constituting five layers is illustrated in Fig. 2. The SSDAE_CS model is a deep neural network consisting of multiple layers of basic SAE and DAE, in which the outputs of each layer are wired

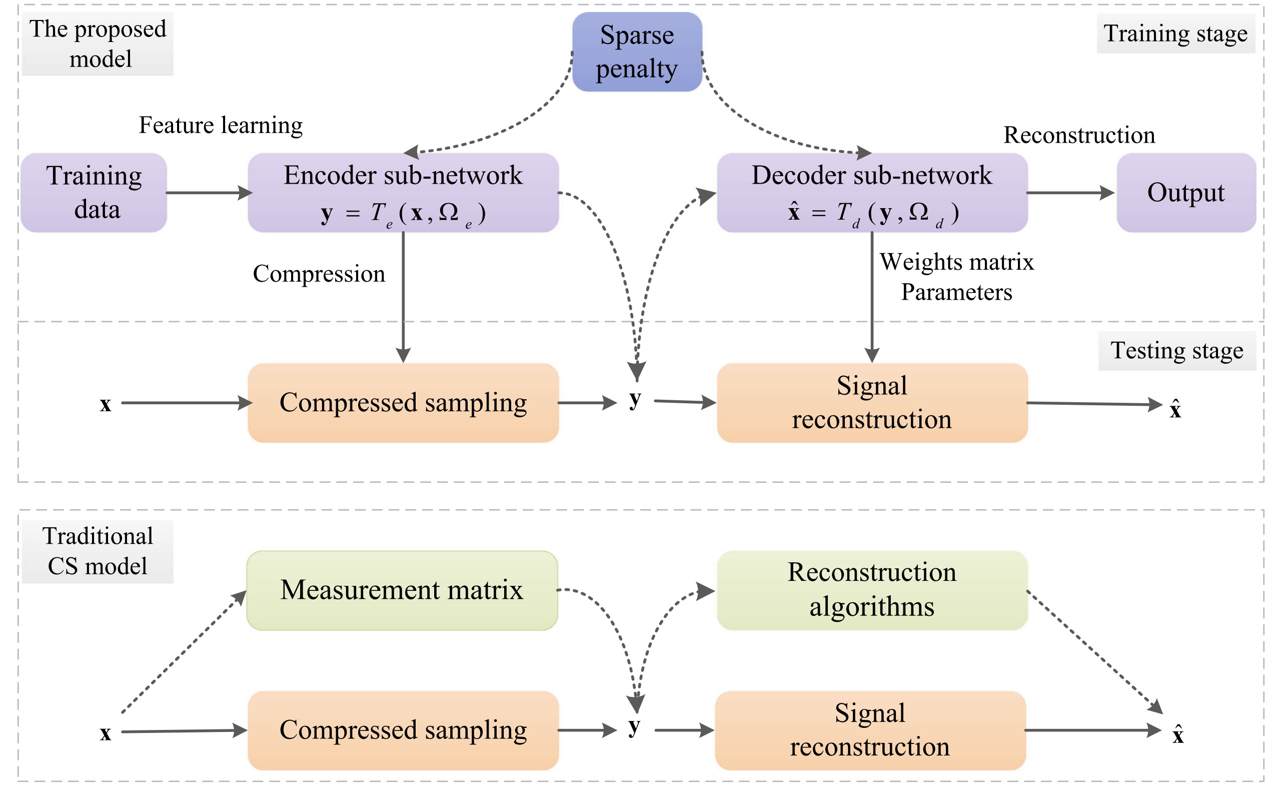

Fig. 1 Comparison between traditional CS model and the proposed model

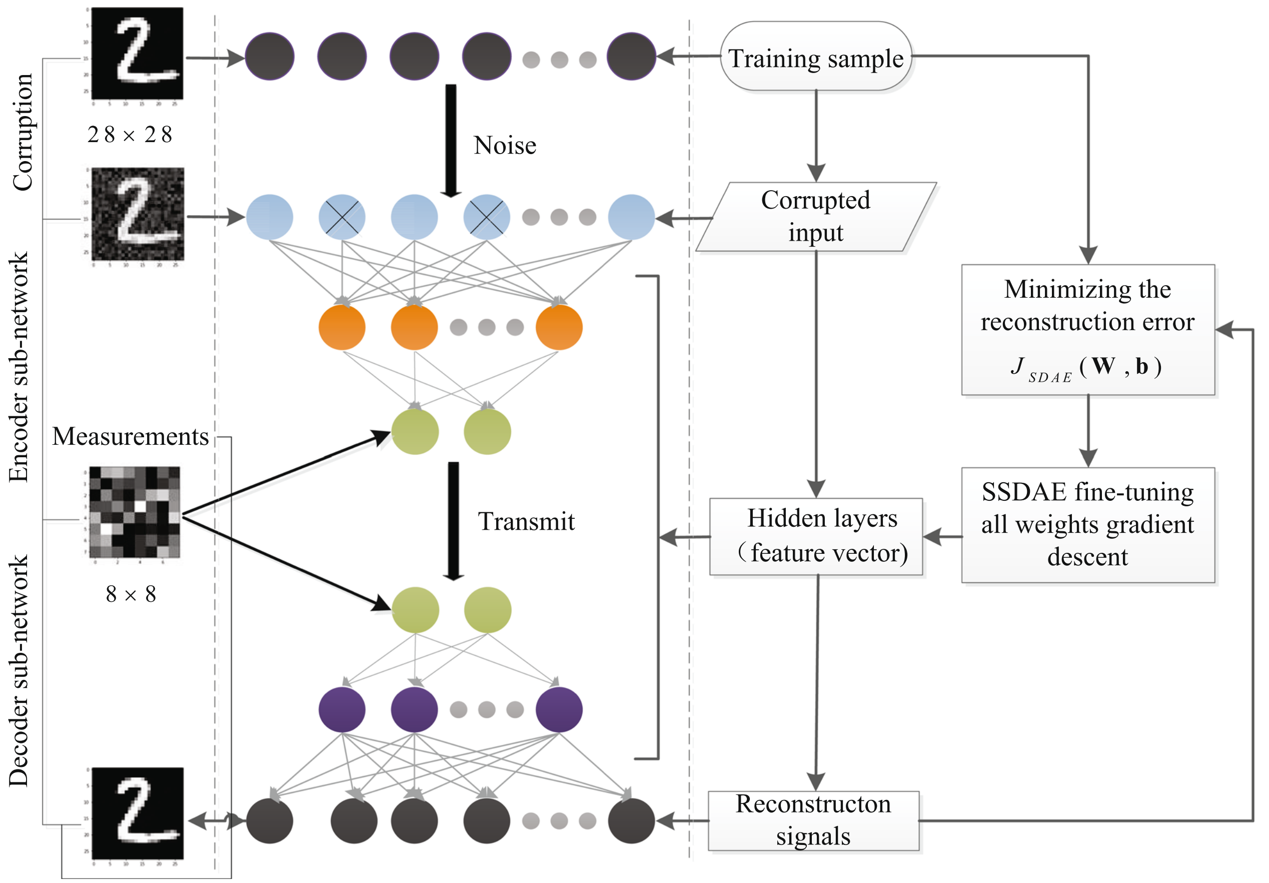
Corruption
Encoder sub-network
Decoder sub-network
28 × 28
Noise
Measurements
8 × 8
Transmit
Training sample
Corrupted input
Minimizing the reconstruction error
$J_{SDAE}(\mathbf{W},\mathbf{b})$
Hidden layers (feature vector)
SSDAE fine-tuning all weights gradient descent
Reconstructon signals

Fig. 2 Architecture of the SSDAE_CS Layer_5 when $M = 64$

to the inputs of each successive layer. It is remarkable that the proposed model is robust to the input because it can reconstruct the original signals from the corrupted input. The proposed model extracts robust features by sparse penalty term, which punishes and inhibits the larger change in the hidden layer. In the corruption stage, the original signals are corrupted by additive white Gaussian noise $\tilde{\mathbf{x}} = \mathbf{x} + \lambda n$, where n denotes the additive Gaussian sampling noise of zero mean and variance one, and λ denotes the degree of the corruption of signals. In the encoder sub-network, the signal can be compressed to M measurements by utilizing multiple nonlinear measurement method. The decoder sub-network reconstructs the original signals from measurements by minimizing the reconstruction error between input and output. Finally, the two sub-networks are integrated into SSDAE_CS model by jointly optimizing parameters to improve the overall performance of CS.

The encoder sub-network can be represented as a deterministic mapping $T_e(\bullet)$, which transforms an input $\mathbf{x} \in \mathbb{R}^{d_x}$ into hidden representation space $\mathbf{y} \in \mathbb{R}^{d_y}$. In the compression process of traditional CS, linear measurement $\mathbf{y} = \Phi\mathbf{x}$ is used, but linear measurements are not optimal. In the SSDAE_CS model, multiple nonlinear measurements are applied to obtain measurements in CS, as shown in the encoding part of Fig. 2. It is found from [25] that nonlinear measurements can preserve more effective information compared to traditional linear measurements. The encoder consists of three layers, (a) an input layer with N nodes, (b) the first hidden layer with K nodes, and (c) the second hidden layer with M nodes, where $N > K > M$. The first hidden feature vector is the value of the first hidden layer, which receives the signals as its input in Eq. (1). The final measurement vector $\mathbf{y}$ is the value of the second hidden layer, which receives the first hidden feature vector as its input in Eq. (2).

$$\mathbf{a}^{(1)} = f\left(\mathbf{z}^{(1)}\right) = f\left(\mathbf{W}^{(1)}\tilde{\mathbf{x}} + \mathbf{b}^{(1)}\right). \tag{1}$$

$$\begin{aligned}\mathbf{y} &= f\left(\mathbf{z}^{(2)}\right) = f\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{b}^{(2)}\right)\\ &= f\left(\mathbf{W}^{(2)}f\left(\mathbf{W}^{(1)}\tilde{\mathbf{x}} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right).\end{aligned} \tag{2}$$

In the SSDAE_CS model, measurements are obtained by two matrix multiplications and nonlinear transformations, so this method is called multiple nonlinear measurement method. Measurement vector $\mathbf{y}$ can also be written as:

$$\mathbf{y} = T_e(\tilde{\mathbf{x}}, \Omega_e), \tag{3}$$

where $\Omega_e = \left\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}; \mathbf{b}^{(1)}, \mathbf{b}^{(2)}\right\}$ denotes the set of encoded parameters and $T_e(\bullet)$ denotes the encoding nonlinear mapping function.

The decoder sub-network is used to map the measurement vector $\mathbf{y}$ back to input space $\mathbf{x} \in \mathbb{R}^{d_x}$ by capturing the feature representation in signal reconstruction process. Among the traditional signal recovery algorithms, each iteration in these greedy or iterative algorithms includes multiple matrix-vector multiplication, which has the computational cost. In this paper, a nonlinear mapping is learned from measurements $\mathbf{y}$ to its original signal $\mathbf{x}$ by training; it just needs two matrix-vector multiplications and two nonlinear mappings. The decoder whose nodes are symmetric with the encoder consists of three layers: input layer with M nodes, the first hidden layer with K nodes, and the second hidden layer with N nodes. The decode function Eqs. (4) and (5) are used to recover the reconstruction signals $\hat{\mathbf{x}}$ from measurement vector $\mathbf{y}$.

$$\mathbf{a}^{(3)} = f\left(\mathbf{z}^{(3)}\right) = f\left(\mathbf{W}^{(3)}\mathbf{y} + \mathbf{b}^{(3)}\right). \tag{4}$$

$$\begin{aligned} \hat{\mathbf{x}} &= f\left(\mathbf{z}^{(4)}\right) = f\left(\mathbf{W}^{(4)}\mathbf{a}^{(3)} + \mathbf{b}^{(4)}\right) \\ &= f\left(\mathbf{W}^{(4)} f\left(\mathbf{W}^{(3)}\mathbf{y} + \mathbf{b}^{(3)}\right) + \mathbf{b}^{(4)}\right). \end{aligned} \tag{5}$$

The reconstruction signals $\hat{\mathbf{x}}$ can also be represented as:

$$\hat{\mathbf{x}} = T_d(\mathbf{y}, \Omega_d), \tag{6}$$

where $\Omega_d = \left\{\mathbf{W}^{(3)}, \mathbf{W}^{(4)}; \mathbf{b}^{(3)}, \mathbf{b}^{(4)}\right\}$ denotes the set of decoded parameters and $T_d(\bullet)$ denotes the decoding nonlinear mapping function.

2.3 Offline training algorithm

Given enough training data, the neural networks can learn to represent arbitrary functions as universal function approximators. The major objective of this training phase is to extract the structural features of signals and learn the nonlinear mapping function of signal reconstruction. Specifically, the encoder and decoder sub-networks are integrated into SSDAE_CS model through end-to-end training for strengthening the connection between the two processes. The parameters will be updated constantly to achieve the optimal training model by reducing the loss function.

The SSDAE_CS model is a typical unsupervised learning model, in which the training set D_{train} has N signals whose label is the same as the sample, i.e., $D_{\text{train}} = \{(x_1, x_1), (x_2, x_2), \cdots, (x_n, x_n)\}$. A trained nonlinear mapping $T_e(\bullet)$ acts as the measurement matrix Φ to obtain the measurements $\mathbf{y}$ from the original signals $\mathbf{x}$, and a trained inverse nonlinear mapping $T_d(\bullet)$ acts as the reconstruction algorithm to recover the reconstruction signals $\hat{\mathbf{x}}$ from $\mathbf{y}$ in the proposed model. To ensure the reconstruction signal $\hat{\mathbf{x}}$ close to the original signals $\mathbf{x}$, the squared error is set as the error function for all data, as shown in Eq. (7):

$$\begin{aligned} J_{SDAE}(\mathbf{W}, \mathbf{b}) = & \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{2}||\hat{x}_i - x_i||^2\right) \\ & + \frac{1}{2}\alpha \sum ||\mathbf{W}||^2 + \beta \sum_{j=1} KL(\rho||\hat{\rho}_j). \end{aligned} \tag{7}$$

To prevent model overfitting, the second term limits the weight parameters $\mathbf{W}$ with L_2 norm as a weight decay term that helps to penalize large weight. α denotes the strength of the penalty term. The third term is a sparse penalty term. $\hat{\rho}_j$ represents the average activation value of the j-th neuron in each batch of training set, β controls the strength of the sparsity penalty term, and ρ denotes the expected activation.

The train goal minimizes $J_{SDAE}(\mathbf{W}, \mathbf{b})$ to update the SSDAE_CS's weights $\mathbf{W}$ and biases $\mathbf{b}$; the detailed training process is shown in Algorithm 1. Firstly, parameters Ω_e and Ω_d are randomly initialized to serve the purpose of symmetry breaking. And then the measurement vector $\mathbf{y}$ and the reconstruction signals $\hat{\mathbf{x}}$ are obtained through the encoder and decoder sub-networks, respectively. Next, the loss function $J_{SDAE}(\mathbf{W}, \mathbf{b})$ is computed by Eq. (7) and

Algorithm 1 Training SSDAE_CS model

```
Input:
    training data $\mathbf{x} = \{x_1, x_2, x_3, \cdots, x_n\}$; the number of measurements $M$;
    the coefficient of noise intensity $\lambda$; sparse factor $\rho$.
Output:
    parameters of encoder $\Omega_e = \left\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}; \mathbf{b}^{(1)}, \mathbf{b}^{(2)}\right\}$;
    measurement vector $\mathbf{y}$;
    parameters of decoder $\Omega_d = \left\{\mathbf{W}^{(3)}, \mathbf{W}^{(4)}; \mathbf{b}^{(3)}, \mathbf{b}^{(4)}\right\}$;
    reconstruction signals $\hat{\mathbf{x}}$.
Randomly initialize $\Omega_e$ and $\Omega_d$.
for $i = 1$ to $k$ do
    Measurement vector $\mathbf{y} = T_e(\tilde{\mathbf{x}}, \Omega_e)$.
    Reconstruction signals $\hat{\mathbf{x}} = T_d(\mathbf{y}, \Omega_d)$.
    Compute loss $J_{SDAE}(\mathbf{W}, \mathbf{b})$.
    Optimize $J_{SDAE}(\mathbf{W}, \mathbf{b})$ by Eqs. (8) and (9).
    Update $\Omega_e$ and $\Omega_d$.
end for
```

batch gradient descent algorithm is performed to compute the gradients and update Ω_e and Ω_d. Each iteration of the gradient descent method updates the parameters $\mathbf{W}$ and $\mathbf{b}$ by Eqs. (8) and (9), respectively.

$$\mathbf{W}_{ij}^{(l)} := \mathbf{W}_{ij}^{(l)} - \alpha \frac{\partial}{\partial \mathbf{W}_{ij}^{(l)}} J_{SDAE}(\mathbf{W}, \mathbf{b}), \tag{8}$$

$$\mathbf{b}_{i}^{(l)} := \mathbf{b}_{i}^{(l)} - \alpha \frac{\partial}{\partial \mathbf{b}_{i}^{(l)}} J_{SDAE}(\mathbf{W}, \mathbf{b}), \tag{9}$$

where α is the learning rate. Computing the partial derivatives is the key thing in this process. The partial derivative is given by the back-propagation algorithm:

$$\frac{\partial}{\partial \mathbf{W}_{ij}^{(l)}} J_{SDAE}(\mathbf{W}, \mathbf{b}) = \frac{1}{n} \sum_{k=1}^{n} \frac{\partial J_{SDAE}(\mathbf{W}, \mathbf{b}; x_k, y_k)}{\partial \mathbf{W}_{ij}^{(l)}} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{a}_j^{(l)} \left[\delta_i^{(l+1)} + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right) f'\left(\mathbf{z}_i^{(l+1)}\right) \right] + \alpha \mathbf{W}_{ij}^{(l)}, \tag{10}$$

$$\frac{\partial}{\partial b_{i}^{(l)}} J_{SDAE}(\mathbf{W}, \mathbf{b}) = \frac{1}{n} \sum_{k=1}^{n} \frac{\partial J_{SDAE}(\mathbf{W}, \mathbf{b}; x_k, y_k)}{\partial \mathbf{b}_{i}^{(l)}} = \frac{1}{n} \sum_{k=1}^{n} \left[\delta_i^{(l+1)} + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right) f'\left(\mathbf{z}_i^{(l+1)}\right) \right], \tag{11}$$

where $\delta_i^{(l)} = \left(\sum_{j=1} \mathbf{W}_{ji}^{(l)} \delta_j^{(l+1)} + \beta \left(-\frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right) \right) f'\left(\mathbf{z}_i^{(l+1)}\right)$ denotes error term of node i in layer l, and n denotes the number of training samples.

3 Results and discussion

In this section, a series of experiments are made to evaluate the performance of the SSDAE_CS model. In the first part, performance indicator is introduced and the MNIST dataset is used for training and testing. Then, the detailed experimental results are given in the final part.

3.1 Dataset and performance indicators

The MNIST dataset, which contains 70,000 grayscale images of handwritten digits of size $N = 28 \times 28$, is employed for the experiments. The dataset is divided into 55,000 samples for training, 5000 samples for validation, and 10,000 samples for testing. K denotes the number of non-zero entries in a grayscale image. It can be seen from Table 1 that the K of most grayscale images is concentrated in the range of 100–200. It shows from average that the number of non-zero items is about 19% of the total in a grayscale image. The handwritten digit images are almost sparse in the spatial domain; therefore, the sparse representation of CS is not necessary or $\Psi = \mathbf{I}$, where $\mathbf{I}$ is the unit matrix.

Peak signal-to-noise ratio (PSNR), which is based on the error between the corresponding pixel points, is often used as a performance indicator to evaluate signal reconstruction quality in the field of image compression. The PSNR is defined as:

$$\text{PSNR(dB)=10log}_{10} \frac{\text{peakval}^2}{\text{MSE}}, \tag{12}$$

where peakval is either specified by the user or taken from the range of the image data type (e.g., for unit image it is 255). Mean square error (MSE) is defined as $\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2$.

3.2 Results analysis

As mentioned earlier, one of the main goals recovers signals from undersampled measurements. To verify the impact of the parameters, a series of experiments are examined for finding the best parameter settings. The motivation of these experiments is to prevent parameter estimation errors from propagating into the reconstruction model during training. Set the number of model layers from $L = 3$ to $L = 9$ for verifying the effect of the number of model layers on the experimental results. The number of neuron nodes per layer is set as follows: the number of nodes in input layer and output layer is 784 (28×28) and the number of measurements ranges from 8 to 512, so the neuron nodes are set to 784, 8–512, 784 in the three-layer network; the neuron nodes are set to 784, 512, 8–512, 512, 784 in the five-layer network, respectively; the neuron nodes are set to 784, 512, 256, 8–512, 256, 512, 784 in the seven-layer network, respectively; and the neuron nodes are set to 784, 512, 512, 256, 8–512, 256, 512, 512, 784 in the nine-layer network, respectively.

Figure 3 reveals that the reconstruction performance is not effectively improved when the number of model layers $L > 5$. This is because the loss function converges

Table 1 The distribution of non-zero entries in the MNIST dataset

K	0–50	51–100	101–150	151–200	201–250	251–784	Total	Average
Training set	75	6661	21,204	20,821	5717	522	55,000	149.93
Validation set	5	607	1918	1905	533	32	5000	149.63
Testing set	13	1163	3770	3942	996	116	10,000	151.12

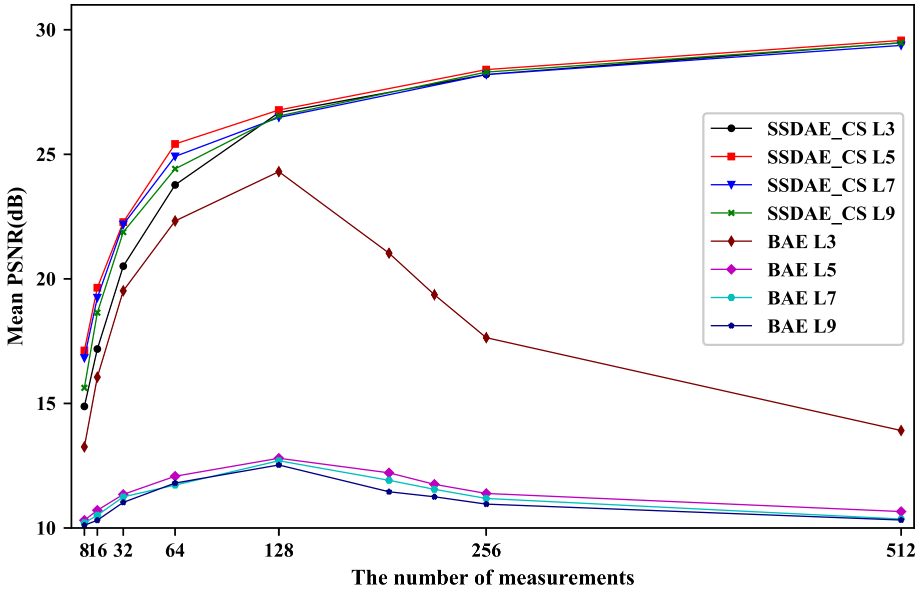

Fig. 3 The experimental results of compared BAE with SSDAE_CS

to a local minimum value rather than a global optimal value as the number of hidden layers increases. Therefore, a SSDAE_CS with five layers is selected as optimal testing model. The mean PSNR of SSDAE_CS is higher than that of basic autoencoder (BAE) without sparsity penalty term, as shown in Fig. 3. And it proves that sparse penalty term improves the model performance by inhibiting the activity of neurons.

To find the optimal sparse factor, different values ρ are tested on SSDAE_CS model with different layers. Figure 4 shows the variation tendency of mean PSNR with sparse factor of hidden units. Experiments show that the model achieves optimal performance when sparse factor $\rho = 0.005$.

Additionally, a series of comparative experiments have been done to evaluate the performance of the proposed

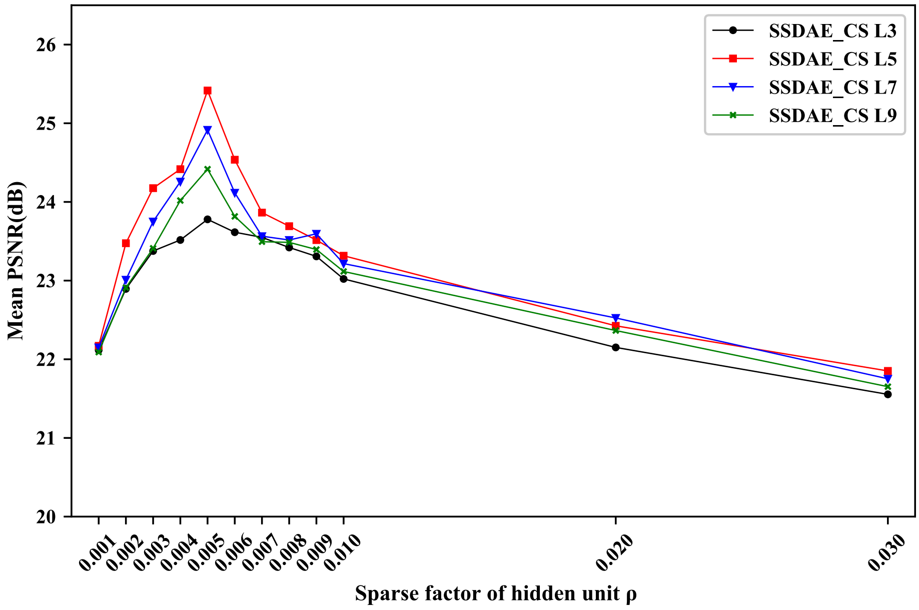

Fig. 4 The variation tendency of mean PSNR with different sparse factor

algorithm in the reconstruction quality and time complexity. Four comparative algorithms are selected from the two categories of reconstruction algorithms: BPDN [17] and OMP [19] based on hand-designed recovery method; DBN-OMP-like and RBM-OMP-like [26] based on data-driven recovery method.

Figure 5 illustrates the variation tendency of mean PSNR of the proposed method and others; it can be clearly seen that the reconstruction performance of the proposed model is significantly better than other algorithms. Firstly, the proposed model not only require fewer measurements than conventional OMP and BPDN to achieve stable recovery but also attain higher mean PSNR values for the entire range of measurements. The main reason for this problem is that the SSDAE_CS model can obtain an optimal function approximator (decoder) by capturing the structural features of training signals in the training process. However, the reconstructed object is a signal in OMP and BPDN, not a batch of signals. Once the signal loses too much information during the compression process, OMP and BPDN cannot reconstruct the signal or the quality of reconstruction is poor. Then, Fig. 5 displays that the reconstruction performance of DBN-OMP-like and RBM-OMP-like, which are based on data-driven recovery method, are significantly better than OMP and BPDN. However, the SSDAE_CS model attains higher PSNR values in the range of measurements $M < 350$ and requires fewer measurements to achieve stable recovery than DBN-OMP-like and RBM-OMP-like. The reconstruction performance of SSDAE_CS is slightly lower than that of DBN-OMP-like and RBM-OMP-like when $M > 350$. The reason for this result is that DBN-OMP-like and RBM-OMP-like use the traditional linear measurement matrix method in the compressed sampling process. Furthermore, they relatively tear the intrinsic relationship between compressed sampling and signal reconstruction. However, the SSDAE_CS model adopts multiple nonlinear measurement methods to preserve more effective information in the compressed sampling process. Through end-to-end training, compressed sampling and signal reconstruction process are perfectly integrated into the proposed model to improve the overall performance of CS.

Table 2 compares the running time of the decoder net of SSDAE_CS with other CS recovery algorithms. The reconstruction time of SSDAE_CS is an order of magnitude faster than OMP and BPND. The main reason for this problem is that SSDAE_CS needs to consider the reconstruction of a batch of signals, but the signal is gradually reconstructed one by one in OMP and BPDN. And the decoder sub-network of SSDAE_CS just needs two matrix-vector multiplications and two nonlinear mappings while other CS recovery algorithms need hundreds of iterations which include multiple matrix-vector multiplications. More precisely, the five-layer SSDAE_CS model requires from 812,824 to 1,329,424 parameters. Although the SSDAE_CS model takes a lot of time to train parameters, the SSDAE_CS model is still very attractive when dealing with large numbers of signals. The SSDAE_CS model spends less time than the RBM-OMP-like and DBN-OMP-like model in the reconstruction process. The reason for this result is that the RBM-OMP-like model requires 4,924,304

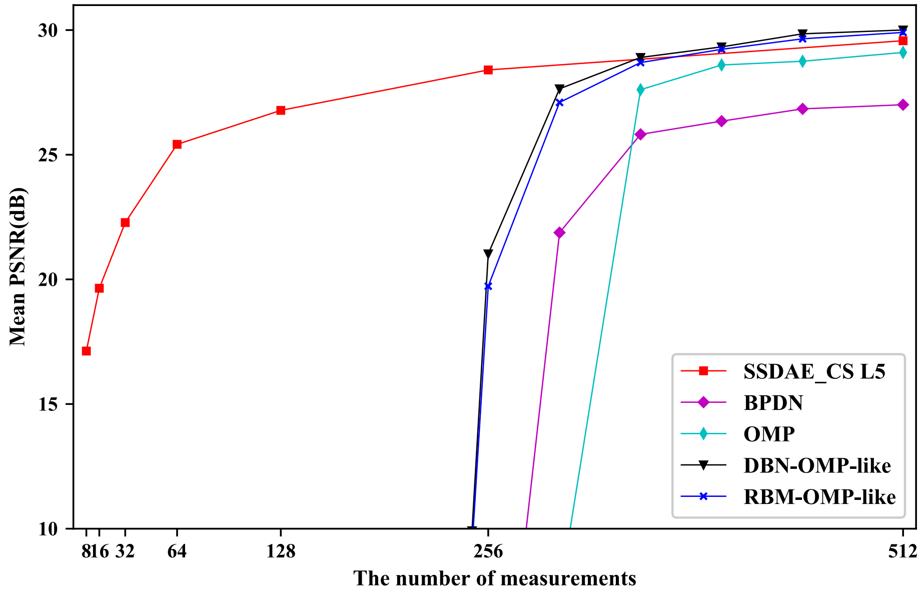

Fig. 5 Evaluation of the reconstruction performance of the MNIST dataset in different reconstruction algorithms

Table 2 Average reconstruction time of MNIST testing set for different *M* and reconstruction algorithms

M		8	1 6	3 2	64	128	256	512
Reconstruction time(s)	OMP	-[a]	-	-	-	-	0.41	0.74
	BPDN	-	-	-	-	-	5.54	4.89
	DBN-OMP-like	-	-	-	-	-	0.03	0.04
	RBM-OMP-like	-	-	-	-	-	0.04	0.04
	SSDAE_CS	0.01	0.01	0.01	0.01	0.01	0.01	0.01

[a] "-" denotes that the signal cannot be recovered or the mean PSNR is less than 10 (dB)

parameters and the DBN-OMP-like model only requires 1,847,104 parameters. RBM-OMP-like, DBN-OMP-like, and SSDAE_CS model use neural networks to learn how to best use the structure within the data, so they are still in the same order of magnitude.

Finally, some experiments have also been made to prove that the proposed model is stable and has a strong denoising ability. In Fig. 6, visual evaluation of a reconstructed test image using the proposed CS model is presented. In the experiments, the white Gaussian noise is added to the training and testing sets and the SSDAE_CS model with different layers are retrained for experimental comparison. Figure 7 shows the tendency of the mean PSNR with the number of measurements in SSDAE_CS model. It can be seen from Fig. 7 that the mean PSNR of the noisy SSDAE_CS model is 3–5 dB lower than the model without noise. To verify the effect of different coefficient noises on the signal reconstruction process of SSDAE_CS model, the comparative experiments are performed when the number of measurements is $M = 64$, and the results are shown in Fig. 8.

4 Conclusions

In this paper, to solve the two most important issues in CS, a SSDAE_CS model has been developed, which contains two sub-networks: an encoder sub-network and a decoder sub-network, respectively, used for compressed sampling and signal recovery. The two sub-networks are integrated into SSDAE_CS model by jointly training parameters to improve the overall performance of CS, but they can be applied to different scenarios as two independent individuals. It is found from simulations that the proposed model requires less the number of measurements to achieve successful reconstruction than other CS reconstruction algorithms, and has a good denoising performance, especially in the case of a few measurements, the performance of the proposed model is better than other methods. In run time of signal reconstructions, the SSDAE_CS model is also

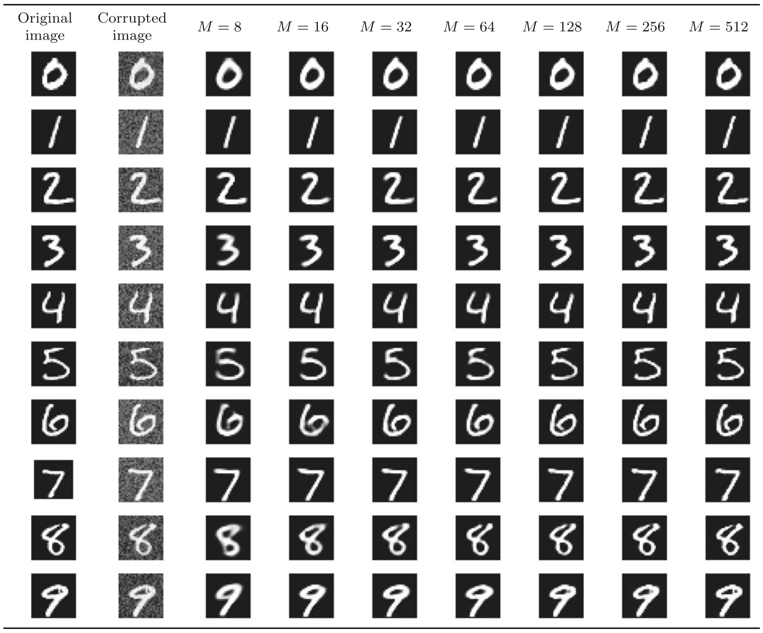

Fig. 6 Visual evaluation of the SSDAE_CS Layer_5 when λ=0.1

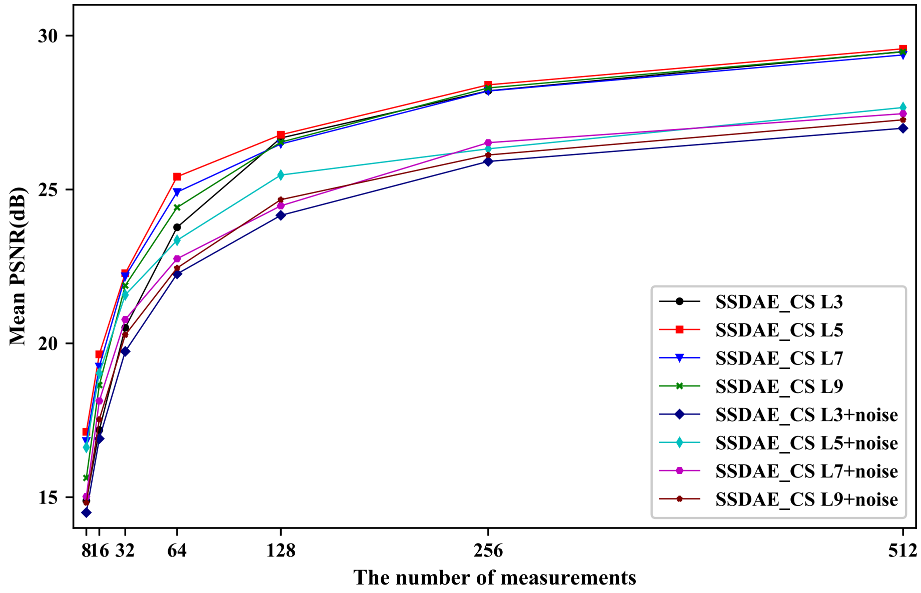

Fig. 7 The variation tendency of mean PSNR in testing set when λ=0.1

faster than other signal recovery algorithms. Considering reconstruction performance, time cost, and denoising ability, the proposed model has a strong attraction for the recovery problems of a large number of signals.

The above paragraph summarizes the advantages of our work, but there are still shortcomings in our work, mainly focusing on accurately reconstructing signals with a few measurements, which requires lots of time and data for training. In further work, transfer learning, which is a convenient alternative for leveraging existing models and updating them on smaller computational platforms and target data sets [33], could be taken into account to address this issue. Additionally, there still exist some compressed sensing problems of big-size nature images; it is worthy to develop convolutional method [34] for sense images, so as to reduce the memory of measurement matrix. Last but not least, residual learning [35] could also be introduced to further increase the depth of network.

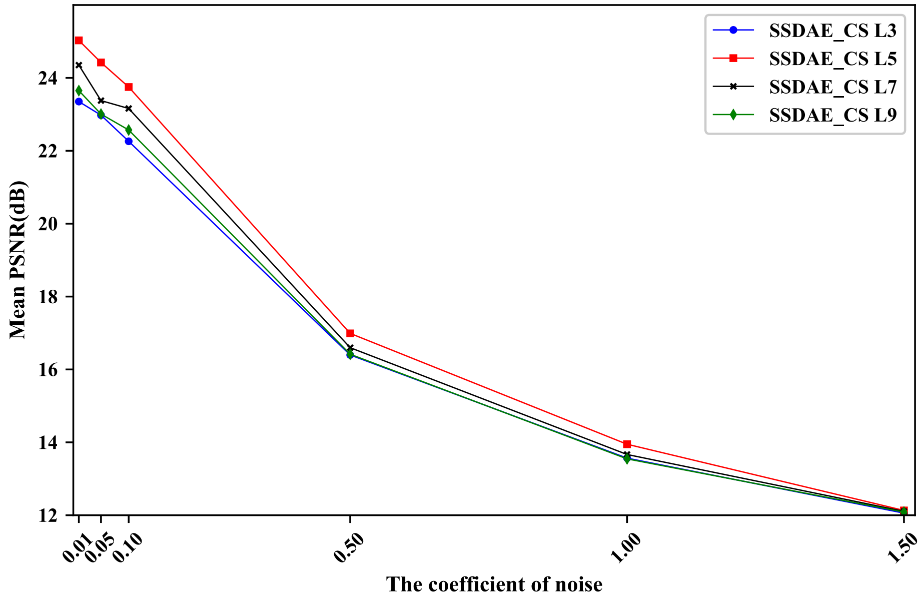

Fig. 8 The mean PSNR for different coefficient in the testing dataset when $M = 64$

Abbreviations

BAE: Basic autoencoder; BCS-BP: Bayesian framework via belief propagation; BCS-LP: Bayesian via laplace prior; BPDN: Basis pursuit denoising; CoSaMP: Compressive sampling matching pursuit; CS: Compressed sensing; DAE: Denoise autoencoder; IHT: Iterative hard thresholding; MSE: Mean square error; OMP: Orthogonal matching pursuit; PSNR: Peak signal-to-noise ratio; SAE: Sparse autoencoder; SSDAE_CS: Stacked sparse denoising autoencoder compressed sensing; TV: Minimization of total variation

Acknowledgements

The authors want to acknowledge the help of all the people who influenced the paper. Specifically, they want to acknowledge the anonymous reviewers for their reasonable comments.

Funding

This work is supported by Natural Science Foundation of China (Grant Nos. 61702066 and 11747125), Chongqing Research Program of Basic Research and Frontier Technology (Grant Nos. cstc2017jcyjAX0256 and cstc2018jcyjAX0154), and Research Innovation Program for Postgraduate of Chongqing (Grant Nos. CYS17217 and CYS18238).

Availability of data and materials

The datasets used during the current study are available from the corresponding author on reasonable request.

Authors' contributions

QZ analyzed the connection between compressed sensing and deep learning. ZZ realized the deduction and design of SSDAE_CS model. YW completed the simulation of the experiments and the analysis of the results , as well as drafting the manuscript. CG checked the manuscript and offered critical suggestions to design the algorithm. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China. [2]School of cyber security and information law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

Received: 6 December 2018 Accepted: 9 April 2019
Published online: 29 April 2019

References

1. D. L. Donoho, Compressed sensing. IEEE Trans. Inf. Theory. **52**(4), 1289–1306 (2006)
2. L. Weizman, Y. C. Eldar, D. B. Bashat, Compressed sensing for longitudinal MRI: an adaptive-weighted approach. Med. Phys. **42**(9), 5195–5208 (2015)
3. Z. Zhang, C. Wang, C. Gan, et al., Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD. IEEE Trans. Sign. Inf. Process. Netw. https://doi.org/10.1109/TSIPN.2019.2900201
4. C. Yan, H. Xie, J. Chen, et al., A fast Uyghur text detector for complex background images. IEEE Trans. Multimed. **20**(12), 3389–3398 (2018)
5. C. Yan, L. Li, C. Zhang, et al., Cross-modality bridging and knowledge transferring for image understanding. IEEE Trans. Multimed. https://doi.org/10.1109/TMM.2019.2903448
6. E. Sejdic, I. Orovic, S. Stankovic, Compressive sensing meets time-frequency: an overview of recent advances in time-frequency processing of sparse signals. Digit. Signal Proc. **77**, 22–35 (2018)
7. Z. Zhang, L. Wang, Y. Zou, et al., The optimally designed dynamic memory networks for targeted sentiment classification. Neurocomputing. **309**, 36–45 (2018)
8. J. Liu, K. Huang, G. Zhang, An efficient distributed compressed sensing algorithm for decentralized sensor network. Sensors. **17**(4), 907 (2017)
9. Z. Zhang, Y. Zou, C. Gan, Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. Neurocomputing. **275**, 1407–1415 (2018)
10. P. Wojtaszczyk, Stability and instance optimality for Gaussian measurements in compressed sensing. Found. Comput. Math. **10**(1), 1–13 (2010)
11. W. Lu, W. Li, K. Kpalma, et al., Compressed sensing performance of random Bernoulli matrices with high compression ratio. IEEE Signal Process. Lett. **22**(8), 1074–1078 (2015)
12. S. Foucart, Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. Springer Proc. Math. **13**, 65–77 (2012)
13. R. A. Devore, Deterministic constructions of compressed sensing matrices. J. Complex. **23**(4-6), 918–925 (2007)
14. W. Yin, Practical compressive sensing with Toeplitz and circulant matrices. Vis. Commun. Image Proc., 77440K (2010). https://doi.org/10.1117/12.863527
15. V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, et al., in *IEEE 18th European Signal Processing Conference*. On optimization of the measurement matrix for compressive sensing, (IEEE, Aalborg, 2010), pp. 427–431
16. X. Gao, J. Zhang, W. Che, et al., in *Data Compression Conference*. Block-based compressive sensing coding of natural images by local structural measurement matrix, (IEEE, Snowbird, 2015), pp. 133–142
17. W. Lu, N. Vaswani, Regularized modified BPDN for noisy sparse reconstruction with partial erroneous support and signal value knowledge. IEEE Trans. Signal Process. **60**(1), 182–196 (2010)
18. C. Li, W. Yin, H. Jiang, et al., An efficient augmented Lagrangian method with applications to total variation minimization. Comput. Optim. Appl. **56**(3), 507–530 (2013)
19. J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory. **53**(12), 4655–4666 (2007)
20. D. Needell, J. A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Commun. ACM. **12**, 93–100 (2010)
21. T. Blumensath, M. E. Davies, Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009)
22. S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing. IEEE Trans. Signal Process. **56**(6), 2346–2356 (2008)
23. D. Baron, S. Sarvotham, R. G. Baraniuk, Bayesian compressive sensing via belief propagation. IEEE Trans Signal Process. **58**(1), 269–280 (2009)
24. C. A. Metzler, A. Maleki, R. G. Baraniuk, From denoising to compressed sensing. IEEE Trans. Inf. Theory. **62**(9), 5117–5144 (2014)
25. A. Mousavi, A. B. Patel, R. G. Baraniuk, in *IEEE Allerton Conference on Communication, Control, and Computing*. A deep learning approach to structured signal recovery, (IEEE, Monticello, 2016), pp. 1336–1343
26. L. Polania, K. Barner, Exploiting restricted Boltzmann machines and deep belief networks in compressed sensing. IEEE Trans. Signal Process. **65**(17), 4538–4550 (2017)
27. A. Mousavi, R. G. Baraniuk, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Learning to invert: signal recovery via deep convolutional networks, (IEEE, New Orleans, 2017), pp. 2272–2276
28. K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, et al., in *IEEE Conference on Computer Vision and Pattern Recognition*. ReconNet: non-iterative reconstruction of images from compressively sensed measurements, (IEEE, Las Vegas, 2016), pp. 449–458
29. P. Vincent, A connection between score matching and denoising autoencoders. Neural Comput. **23**(7), 1661–1674 (2011)
30. P. Xiong, H. Wang, M. Liu, et al., A stacked contractive denoising auto-encoder for ECG signal denoising. Physiol. Meas. **37**(12), 2214–2230 (2016)
31. A. Lemme, R. F. Reinhart, J. J. Steil, Online learning and generalization of parts-based image representations by non-negative sparse autoencoders. Neural Netw. **33**(9), 194–203 (2012)
32. J. Xu, L. Xiang, Q. Liu, et al., Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE Trans. Med. Imaging. **35**(1), 119–130 (2016)

33. D. Xu, D. Lan, H. Liu, et al., Compressive sensing of stepped-frequency radar based on transfer learning. IEEE Trans. Signal Process. **63**(12), 3076–3087 (2015)
34. J. Du, X. Xie, C. Wang, et al., Fully convolutional measurement network for compressive sensing image reconstruction. Neurocomputing. **328**, 105–112 (2019)
35. K. He, X. Zhang, S. Ren, et al., in *2016 IEEE Conference on Computer Vision and Pattern Recognition*. Deep residual learning for image recognition, (IEEE, Las Vegas, 2016), pp. 770–778