Auto-encoders for compressed sensing

Pei Peng
Rutgers University
Piscataway, NJ 08854
pp566@scarletmail.rutgers.edu

Shirin Jalali, Xin Yuan
Bell Labs, Nokia
Murray Hill, NJ 07974
{shirin.jalali,xin_x.yuan}@nokia-bell-labs.com

Abstract

Compressed sensing is about recovering a structured high-dimensional signal $\mathbf{x} \in \mathbb{R}^n$ from its under-determined noisy linear measurements $\mathbf{y} \in \mathbb{R}^m$, where $m \ll n$. While the vast majority of the literature in this area is on sparse signals, in recent years, there has been considerable progress on compressed sensing of signals with structures beyond sparsity. One of the promising approaches in this field is to employ generative models that are based on trained neural networks. In this paper, we study the performance of an iterative algorithm based on projected gradient descent that employs an auto-encoder to define and enforce the source structure. The auto-encoder is defined by a generative function $g : \mathbb{R}^k \to \mathbb{R}^n$ and a separate neural network that is trained to function as the inverse of g. We prove that, for a generative model g with ℓ_2 representation error δ, given roughly $m > 40k \log \frac{1}{\delta}$ measurements, such an algorithm converges, even in the presence of additive white Gaussian noise.

1 Problem statement

Compressed sensing (CS), i.e., recovering a structured signal $\mathbf{x} \in \mathbb{R}^n$ from its under-determined noisy linear measurements $\mathbf{y} \in \mathbb{R}^m$, where $\mathbf{y} = A\mathbf{x} + \mathbf{z}$, and $m \ll n$, has been an active area of research during the past decade. (Here, $A \in \mathbb{R}^{m \times n}$ and $\mathbf{z} \in \mathbb{R}^m$ denote the sensing matrix and the measurement noise, respectively.) While the majority of initial work in this area focused on sparse signals, the class of structures employed by recovery methods has steadily grown to include more complex structures as well. Moving beyond simple structures and taking advantage of structures beyond sparsity can lead to better performance in terms of the required number of measurements and reconstruction quality. However, designing algorithms that employ other more complex structures is a challenging problem. Generally, for a given class of signals $\mathcal{Q}$ with unknown underlying structure both i) discovering all existing structures in $\mathcal{Q}$, and ii) designing an efficient CS recovery algorithm that employs the discovered structure are challenging. To address these issues, in recent years, researchers have approached this problem from different perspectives.

One of the appealing novel approaches in this area is to use generative models (GMs) based on trained neural networks (NNs). The well-known universal approximation theory (UAT) states that with proper weights, NNs can approximate any regular function with arbitrary precision [1, 2, 3, 4]. This suggests that trained NNs operating as generative functions are potentially capable of capturing complex unknown structures.

In this paper, we focus on GMs that are based on NNs and explore application of such models in designing efficient and robust CS recovery methods. Consider a class of structured signals $\mathcal{Q}$, a compact subset of $\mathbb{R}^n$. Further, consider a GM $g : \mathcal{U}^k \to \mathbb{R}^n, \mathcal{U} \subset \mathbb{R}$, such that g is able to represent signals from $\mathcal{Q}$ closely. In this work, we study, both theoretically and empirically, the performance of an iterative CS recovery method based on projected gradient descent (PGD) that employs g to capture and enforce the source model (structure). We connect the number of measurements required by such a recovery method with the properties of function g.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Related work. Using NNs for CS has been an active area of research in recent years. (See [5, 6, 7, 8, 9, 10, 11, 12, 13] for a non-comprehensive list of such results.) Closely studying the literature in this area reveals that, interestingly, the role imagined for the NN to play is not shared among different approaches. While in some methods, NNs are directly trained to solve the inverse problem, in others, they are trained, independent of the CS recovery problem, as GMs that capture the source model. Our focus in this paper is on the latter type of methods where the role of the NN is to build a powerful GM that captures the source complex structure. Application of NN-based GMs to solve CS problems was first proposed in [9]. In [12], an iterative algorithm based on PGD (similar to the one studied here) was proposed and studied. Here, we derive sharp theoretical guarantees for the PGD-based algorithm. Our bounds directly connect the number of measurements with the properties of the GM, such as its input dimension and its representation quality.

Notations. Vectors are denoted by bold letters, such as $\mathbf{x}$. Sets are denoted by calligraphic letters, such as $\mathcal{A}$. For a set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality. $\log$ and $\ln$ refer to logarithm in base 2 and normal logarithm, respectively.

2 GM-based recovery

Consider a class of signals defined by a compact set $\mathcal{Q} \subset \mathbb{R}^n$. Let function $g : \mathcal{U}^k \to \mathbb{R}^n$ denote a generative function (model) trained to represent signals in set $\mathcal{Q}$. Assume that function g is L-Lipschitz, and $\mathcal{U}$ is a bounded subset of $\mathbb{R}$.

Definition 1 *Function $g : \mathcal{U}^k \to \mathbb{R}^n$ is said to cover set $\mathcal{Q}$ with distortion δ, if*

$$\sup_{\mathbf{x}\in\mathcal{Q}} \min_{\mathbf{u}\in\mathcal{U}^k} \tfrac{1}{\sqrt{n}}\|g(\mathbf{u}) - \mathbf{x}\| \leq \delta. \tag{1}$$

Consider the standard problem of CS, where signal $\mathbf{x} \in \mathcal{Q}$ is measured as $\mathbf{y} = A\mathbf{x} + \mathbf{z}$ with $A \in \mathbb{R}^{m\times n}$, $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{z} \in \mathbb{R}^m$ denoting the sensing matrix, measurement vector, and measurement noise, respectively. Instead of knowing the structure of signals in $\mathcal{Q}$, assume that we have access to the GM g, which is able to represent signals in $\mathcal{Q}$ with high fidelity. To solve this problem, ideally, one needs to find a signal that is i) compatible with measurements $\mathbf{y}$, and ii) representable with function g. Hence, ignoring the computational complexity, we would like to solve the following optimization problem proposed in [9]:

$$\hat{\boldsymbol{u}} = \operatorname{argmin}_{\mathbf{u}\in\mathcal{U}^k} \|Ag(\mathbf{u}) - \mathbf{y}\|. \tag{2}$$

After finding $\hat{\boldsymbol{u}}$, signal $\mathbf{x}$ can be recovered as

$$\hat{\boldsymbol{x}} = g(\hat{\boldsymbol{u}}). \tag{3}$$

Using the described optimization, [9] proves that roughly $O(k \log L)$ measurements are sufficient for recovering $\mathbf{x}$ from $\mathbf{y}$. Furthermore, in the case of $\delta = 0$, i.e., the case where the GM has no representation error, [14] proves that slightly more than $2k$ noiseless measurements are sufficient for almost lossless recovery.

3 PGD-based algorithm

The optimization described in (2) is a challenging non-convex optimization. The GM g defined using an NN is a differentiable function. Therefore, [9] suggests solving $\min_{\mathbf{u}\in\mathcal{U}^k} \|Ag(\mathbf{u}) - \mathbf{y}\|$ through applying the standard gradient descent (GD) algorithm. However, since the problem is non-convex, there is no guarantee that the solution derived based on this approach is close to the optimal solution. Another approach is to apply PGD as follows: For $t = 0, 1, \ldots$, let

$$\mathbf{s}^{t+1} = \hat{\boldsymbol{x}}^t + \mu A^T(\mathbf{y} - A\hat{\boldsymbol{x}}^t)$$

$$\mathbf{u}^{t+1} = \operatorname{argmin}_{\mathbf{u}\in\mathcal{U}^k} \|\mathbf{s}^{t+1} - g(\mathbf{u})\| \tag{4}$$

$$\hat{\boldsymbol{x}}^{t+1} = g(\mathbf{u}^{t+1}). \tag{5}$$

Consider $\mathbf{x} \in \mathcal{Q}$. Assume that the described PGD-based method is used recovering $\mathbf{x}$ from m-dimensional noisy measurements $\mathbf{y} = A\mathbf{x} + \mathbf{z}$. The following theorem connects the number of measurements m, the representation error of the GM δ, the input dimension of the GM k, with the convergence performance of the PGD-based algorithm. Furthermore, it shows the robustness of this approach to additive white Gaussian noise.

Theorem 1 *Consider* $\mathbf{x} \in \mathcal{Q}$*, and let* $\mathbf{y} = A\mathbf{x} + \mathbf{z}$*. Assume that the entries of* A *and* $\mathbf{z}$ *are i.i.d.* $\mathcal{N}(0,1)$ *and i.i.d.* $\mathcal{N}(0,\sigma^2)$*, respectively. Assume that function* $g : [0,1]^k \to \mathbb{R}^n$ *is L-Lipschitz and satisfies* (1)*, for some* $\delta > 0$*. Define* $\tilde{\mathbf{u}}$ *and* $\tilde{\boldsymbol{x}}$ *as* $\operatorname{argmin}_{\mathbf{u}\in\mathcal{U}^k} \|\mathbf{x} - g(\mathbf{u})\|$ *and* $\tilde{\boldsymbol{x}} = g(\tilde{\mathbf{u}})$*, respectively. Choose free parameters* $\alpha, \upsilon \in \mathbb{R}^+$ *and define* $\eta \triangleq \frac{k}{n}(1 + (\sqrt{\frac{n}{m}} + 2)^2)L^2\delta^{2\alpha}$*,* $\gamma_1 \triangleq (2 + \sqrt{\frac{n}{m}})^2(L\delta^\alpha\sqrt{\frac{k}{n}} + 1)$ *and* $\gamma_2 \triangleq \sqrt{\frac{2k}{n}}(2 + \sqrt{\frac{n}{m}})$*. Assume that*

$$m \geq 40(1 + \alpha + \upsilon)k \log \tfrac{1}{\delta}. \tag{6}$$

Let $\mu = \frac{1}{m}$*. For* $t = 0, 1, \ldots$*, define* $(\mathbf{s}^{t+1}, \mathbf{u}^{t+1}, \hat{\boldsymbol{x}}^{t+1})$ *as* (5)*. Then, for every* t*, if* $\frac{1}{\sqrt{n}}\|\hat{\boldsymbol{x}}^t - \tilde{\boldsymbol{x}}\| \geq \delta$*, then, either* $\frac{1}{\sqrt{n}}\|\hat{\boldsymbol{x}}^{t+1} - \tilde{\boldsymbol{x}}\| \leq \delta$*, or*

$$\tfrac{1}{\sqrt{n}}\|\tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}^{t+1}\| \;\leq\; \tfrac{0.9+\eta}{\sqrt{n}}\|\tilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}^t\| + \left(\sqrt{\tfrac{6(1+\alpha)\left(\log\frac{1}{\delta}\right)k}{m}} + \gamma_2 L\delta^\alpha\right)\tfrac{\sigma}{\sqrt{n}} + \gamma_1\delta,$$

with a probability larger than $1 - 2^{-2k\upsilon\log\frac{1}{\delta}} - \mathrm{e}^{-\frac{m}{2}} - \mathrm{e}^{-0.1(1+\alpha)\left(\log\frac{1}{\delta}\right)k + 2(\ln 2)k} - \mathrm{e}^{-0.15m}$.

Theorem 1 states that although the original optimization is not convex, having roughly more than $40k\log\frac{1}{\delta}$ measurements, the described PGD algorithm converges, even in the presence of additive white Gaussian noise.

In order to implement the proposed iterative method described in (5), the step that might seem challenging is the projection step, i.e., $\mathbf{u}^{t+1} = \operatorname{argmin}_{\mathbf{u}\in\mathcal{U}^k}\|\mathbf{s}^{t+1} - g(\mathbf{u})\|$. Since the cost function is differentiable , one can use GD to solve it [12]. However, since the cost is not convex, there is no guarantee that the solution will be close to the optimal. Moreover, using GD to solve this optimization, adds to the computational complexity of the problem. Therefore, instead, we consider training a separate neural network that approximates the solution to this optimization. Concatenating this neural network with the the neural network that define g essentially yields an "auto-encoder" (AE) that maps a high-dimensional signal into low-dimensions, and then back to its original dimension. Using this perspective, the last two steps of the algorithm, basically pass $\mathbf{s}^{t+1}$ through an AE. (See Fig. 1.)

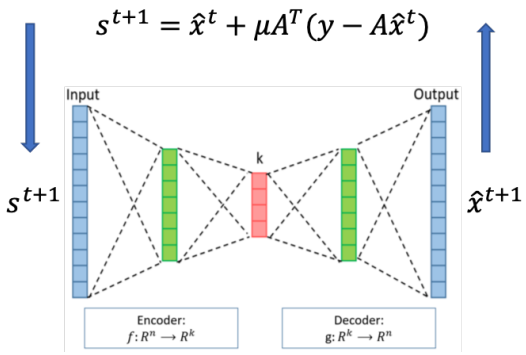

Figure 1: AE-based CS recovery

4 Simulation results

We conduct simulations to study the performance of the AE-based CS algorithm. The AE structure (2-layer encoder, and 2-layer decoder) and the PGD algorithm are both shown in Fig. 1. The implementations are done in PyTorch using Nvidia 1080 Ti GPU. We study both the MNIST hand-written digits [15] and the chest X-ray images provided by NIH [16]. In both cases, the average peak signal-to-noise ratio (PSNR) is used to evaluate the quality of the reconstructed images.

MNIST In our first set of experiments we study the MNIST dataset. Each MNIST image is 28×28. We use 35000 images for training and 300 images for testing. We consider an AE with fully-connected layers, such that the hidden layers of the encoder and the decoder (GM) each consists of 1500 hidden nodes. Also, $k = 100$, i.e., the size of the output layer of the encoder and the input size of the GM. To apply the iterative AE-based algorithm, we set $\mu = 0.7$.

Fig. 2 compares the performance of the AE-based recovery with Lasso [17] and BM3D-AMP [18] under different sample rates m/n, in both noise-free and noisy settings (middle plot with signal-to-noise-ratio (SNR) of 10 dB). It can be seen that in the *noise free* case, when the sampling rate is low (e.g. 0.1 and 0.05), the AE-based method outperforms the other methods. When the sampling rate is higher (e.g. 0.2 and 0.3), BM3D-AMP achieves the best performance. In the *noisy* case, although BM3D-AMP still has the highest PSNR in high sample rate cases, its performance drops significantly. Some reconstructed images by the three algorithms (under noise free case) compared with the ground truth are shown in the right plot.

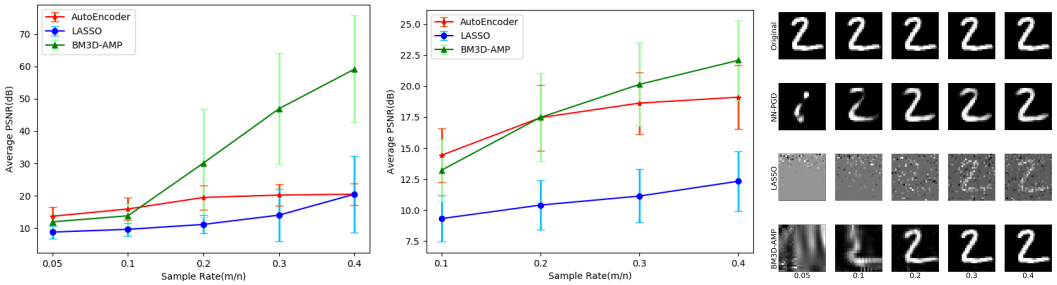

Figure 2: Comparing Lasso, BM3D-AMP and the proposed auto-encoder based inversion in the *noise free* case (left) and *noisy* case (middle with SNR = 10dB). Right: reconstructed images at different sampling rate in the noise free case.

X-ray Images We next explore the performance of the AE-based method when applied to the chest X-ray images. In this dataset, each image is 128×128, and there are 35000 training images and 100 testing images. We compare the performance of the AE-based method with BM3D-AMP and Lasso-DCT. We consider two different NNs calling the results NN1-PGD and NN2-PGD. Both NNs are structured as before with different number of nodes: i) for NN1, $k = 200$ and there are 5000 hidden nodes in the first layer of the encoder and the second layer of the decoder, and ii) for NN2, $k = 300$, there are hidden 8000 hidden nodes in the first layer of the encoder and the second layer of the decoder.

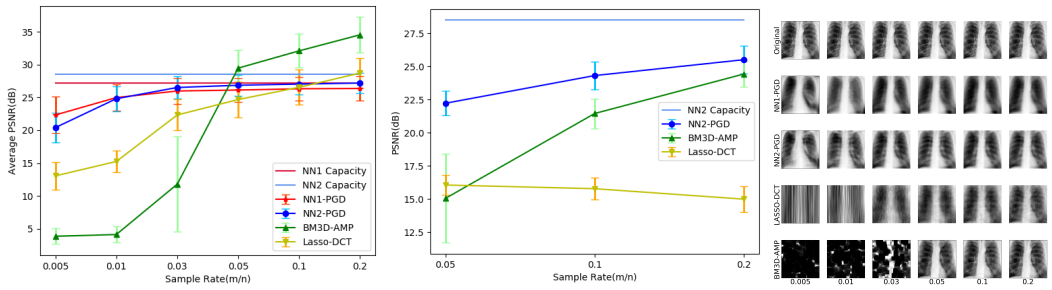

Figure 3: PSNR of the reconstructed X-ray images under noise free (left) and noisy (middle with SNR = 10dB) cases and some example images (right).

Fig. 3 plots the average PSNR on test images in both noiseless (left) and noisy (middle) settings, again at SNR = 10dB. The capacities of the NNs refers to the average representation error corresponding to each NN. Of course, the performance of the AE-based cannot exceed the capacity of the corresponding NN. It can be observed that for both NNs, the AE-based method in fact achieves the capacity. This implies that to achieve better performance by using the AE-based method, we need to design a NN with higher capacity, i.e., lower representation error.

References

[1] G. Cybenko. Approximations by superpositions of a sigmoidal function. *Math. of Cont., Sig. and Sys.*, 2:183–192, 1989.

[2] K.I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neu. net.*, 2(3):183–192, 1989.

[3] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neu. Net.*, 2(5):359–366, 1989.

[4] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Mach. lear.*, 14(1):115–133, 1994.

[5] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. In *Ann. Allerton Conf. on Commun., Cont., and Comp.*, pages 1336–1343. IEEE, 2015.

[6] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proc. of the IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR)*, pages 449–458, 2016.

[7] J.H. Rick Chang, C.-L. Li, B. Poczos, B.V.K. Vijaya Kumar, and A. C. Sankaranarayanan. One network to solve them all- solving linear inverse problems using deep projection models. In *Proc. of the IEEE Int. Conf. on Comp. Vis.*, pages 5888–5897, 2017.

[8] M. Borgerding, P. Schniter, and S. Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Trans. Sig. Proc.*, 65(16):4293–4308, 2017.

[9] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. In *Int. Conf. Mach. Learn.*, pages 537–546, 2017.

[10] X. Yuan and Y. Pu. Parallel lensless compressive imaging via deep convolutional neural networks. *Optics Express*, 26(2):1962–1977, Jan 2018.

[11] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *Trans. Image Proc.*, 26(9):4509–4522, Sep. 2017.

[12] V. Shah and C. Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *Int. Conf. on Aco., Speech, and Sig. Proc. (ICASSP)*, Apr. 2018.

[13] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis. Compressed sensing with deep image prior and learned regularization. *arXiv preprint arXiv:1806.06438*, 2018.

[14] S. Jalali and X. Yuan. Solving linear inverse problems using generative models. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 512–516, 2019.

[15] Y LeCun, C. Cortes, and C. J.C. Burges. The mnist database of handwritten digits. In *http://yann.lecun.com/exdb/mnist/index.html*.

[16] National Institutes of Health Chest X-Ray Dataset. In *https://www.kaggle.com/nih-chest-xrays*.

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Roy. Stat. Soc.: Ser. B (Meth.)*, 58(1):267–288, 1996.

[18] C. A. Metzler, A. Maleki, and R. G. Baraniuk. BM3D-AMP: A new image recovery algorithm based on BM3D denoising. In *2015 IEEE Int. Conf. on Image Proc. (ICIP)*, pages 3116–3120. IEEE, 2015.