Phase Retrieval using Untrained Neural Network Priors

Gauri Jagatap
Electrical and Computer Engineering
Iowa State University
gauri@iastate.edu

Chinmay Hegde
Electrical and Computer Engineering
Iowa State University
chinmay@iastate.edu

Abstract

Untrained deep neural networks as image priors have been recently introduced for linear inverse imaging problems such as denoising, super-resolution, inpainting, and compressive sensing, with promising performance gains over hand-crafted image priors such as sparsity. Moreover, unlike learned generative priors they do not require any training over large datasets. In this paper, we consider the non-linear inverse problem of compressive phase retrieval (CPR); this involves reconstructing a d-dimensional image signal from n magnitude-only measurements, where $n \ll d$. To this end, we model images to lie in the range of an untrained deep generative network with a fixed seed. We then present two approaches for solving CPR — gradient descent, and projected gradient descent — and show superior empirical performance when compared to algorithms that use hand crafted priors.

1 Introduction

Motivation: Low-light images are images that are captured in poor lighting conditions. As an artifact of camera sensors hardware and photon noise, this can result in noisy images with low dynamic resolution. This motivates the low-light image enhancement problem, which is particularly challenging. Several low-light enhancement problems suffer from over-enhancement, where images appear to be unnaturally bright. Mathematically, low-light image enhancement is a non-linear problem; each pixel value is modified by different amounts.

Deep neural networks have led to unprecedented success in solving several linear as well as non-linear inverse imaging problems. Image denoising [1], super-resolution [2], inpainting, compressed sensing [3] and phase retrieval [4] are among the many imaging applications that have benefited from the usage of deep convolutional networks (CNNs). Learned generative priors have replaced hand-crafted priors such as sparsity, total variation, block-matching with high success rates [3, 5, 6, 7]. However, the main challenge with these approaches is the requirement of massive amounts of training data.

In contrast, recently, there has been a surge of interest in using *untrained* neural networks as an image priors, for regularizing linear inverse problems. Seminal works on Deep Image Prior [8, 9] are capable of solving linear inverse imaging problems with no training data whatsoever, while merely imposing an auto-encoder [8] and decoder [9] architecture as a structural priors on the image, for denoising, inpainting, super-resolution and compressed sensing [10] and exhibit superior reconstruction performance as compared to conventional methodologies [11, 12, 13].

Phase retrieval is a non-linear inverse imaging problem which arises in several Fourier imaging applications, and involves reconstructing images from magnitude-only measurements. Given a d-dimensional signal x^*, and measurement matrix A, one can obtain n measurements stored in y, such that the forward model representing overall image acquisition can be expressed as:

$$y = |Ax^*|$$

Workshop on Solving Inverse Problems with Deep Networks, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

where $|\cdot|$ represents an element-wise absolute value operation. The task is to recover signal estimate $\hat{x}$ from y, which can be posed as a non-convex problem of the following form:

$$\hat{x} = \arg\min_{x} \| y - |Ax| \|_2^2.$$

If the measurement matrix A has Gaussian $\mathcal{N}(0, 1/n)$ entries with zero mean and variance $1/n$, then literature suggests that the sample requirement exceeds $n > 2d$ [14] and in high dimensional regimes this can be especially challenging, since the computational complexity is also proportional to n and d.

Compressive phase retrieval (CPR) models use sparsity as a prior for reducing sample requirements; however, standard techniques from recent literature [15] suggest a quadratic dependence of number of measurements on the sparsity level for recovering sparse images from magnitude-only Gaussian measurements and the design of a smart initialization scheme [16, 15], for linear convergence algorithms. Convex relaxation based methods advocate a *lifting* [17] strategy that converts the problem to a rank-one matrix estimation problem which is computationally expensive. Recent papers such as [6, 18] have used learned GAN priors for bringing down sample requirements; however this requires sufficient training data, which can be prohibitively expensive to obtain in domains which require phase retrieval, such as medical or astronomical imaging.

Our contributions: In this paper, we explore the use of untrained deep neural networks as an image prior for inverting images from under-sampled non-linear measurements. We assume that the image x^* belongs to the range spanned by the weights of a deep *under-parameterized* untrained neural network $G(\mathbf{w}; z)$ [9], which we denote by $\mathcal{S}$, where $\mathbf{w}$ is a set of the weights of the deep network, z is the latent code, and G captures the overall deep network architecture. Specifically, we use two different strategies to solve compressive phase retrieval with fewer measurements as compared to state-of-art, based on gradient descent (Net-GD) and projected gradient descent (Net-PGD) and compare these two schemes. We empirically show significant improvements in image reconstruction quality as compared to prior works. We further analyze random initialization schemes in the context of both algorithms.

Prior work: Sparse signal recovery from magnitude-only compressive measurements has been extensively studied, with several algorithmic approaches [15]. In all of these approaches, modeling the low-dimensional embedding is challenging and may not be captured correctly using simple hand-crafted priors such as structured sparsity [19]. Since it is hard to estimate these hyper-parameters accurately, the number of samples required to reconstruct the image is often much higher than information theoretic limits [19, 5]. Several papers in recent literature [14, 20, 15, 16] rely on the design of a spectral initialization scheme which ensures that one can subsequently optimize over a convex ball of the problem. However this initialization requirement results in high sample complexity and is a bottleneck in achieving information theoretically optimal sample complexity. Trained generative priors have successfully cirvumvented this problem [5, 6, 21, 22, 7]; however, with the following shortcomings: test images are strictly restricted to the range of a trained generator, and the requirement of sufficient training data.

Notation: Throughout the paper, lower case letters denote vectors, such as v and upper case letters for matrices, such as M. Vectorization of a matrix is written as $\text{vec}(\cdot)$. Hadamard or element-wise product is denoted by $\circ$. Element-wise absolute valued vector is denoted by $|v|$.

2 Problem setup

In this paper we discuss the problem of reconstructing image $\hat{x}$, with assumed underlying deep image prior, from magnitude-only Gaussian measurements y, which can be formulated as an optimization problem of the form:

$$\min_{x \in \mathcal{S}} \mathcal{L}_1(x) := \min_{x \in \mathcal{S}} \| y - |Ax| \|_2^2 \tag{1}$$

where we have chosen the ℓ_2-squared loss function and where $\mathcal{S}$ captures the prior on the image.

In general, if the image $x \in \mathbb{R}^{d \times k}$ (k channels each of dimension d) can be represented as the action of a deep generative network $G(\mathbf{w}; z)$ with weights $\mathbf{w}$ on some latent code z, such that $x = G(\mathbf{w}; z)$, then the set $\mathcal{S}$ captures the characteristics of $G(\mathbf{w}; z)$. The latent code $z := \text{vec}(Z_1)$ with $Z_1 \in \mathbb{R}^{d_1 \times k_1}$ is a low-dimensional embedding with dimension $d_1 k_1 \ll dk$ and its elements are generated from appropriately scaled uniform random distribution.

Algorithm 1 Net-GD for compressive phase retrieval.

1: **Input:** $A, z = \text{vec}(Z_1), \eta, \mathbf{w}^0$.
2: **while** termination condition not met **do**
3: $\quad \mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta \nabla \mathcal{L}_2(\mathbf{w}^t)$ {gradient step}
4: **end while**
5: **Output** $\hat{x} \leftarrow G(\mathbf{w}^T; z)$.

In this paper, the weights of the generator $\mathbf{w}$ are *not pre-trained*; rather, the task is to estimate image $\hat{x} = G(\hat{\mathbf{w}}; z) \approx G(\mathbf{w}^*; z) = x^*$ and corresponding weights $\hat{\mathbf{w}}$, for a *fixed* seed z, where x^* is assumed to be the true image and the true weights $\mathbf{w}^*$ (possibly non-unique) satisfy $\mathbf{w}^* = \min_{\mathbf{w}} \|x^* - G(\mathbf{w}; z)\|_2^2$. With this formulation, the optimization in Eq. 1 can be rewritten by substituting the surjective mapping $G : \mathbf{w} \to x$, and optimizing over $\mathbf{w}$,

$$\min_{\mathbf{w}} \mathcal{L}_2(\mathbf{w}) := \min_{\mathbf{w}} \|y - |AG(\mathbf{w}; z)|\|_2^2, \tag{2}$$

to obtain $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathcal{L}_2(\mathbf{w})$ and corresponding image $\hat{x}$.

Specifically, the untrained network $G(\mathbf{w}; z)$ takes the form of an expansive neural network; a decoder architecture similar to the one in [9]. Alternatively, one may assume the architecture of the generator of a DCGAN [23, 10]. The neural network is composed of L weight layers W_l, indexed by $l \in \{1, \ldots, L\}$ and are 1×1 convolutions, bilinear upsampling operators U_l for $l \in \{1, \ldots L-1\}$ and ReLU activation $\sigma(\cdot)$ and is expressed as follows

$$x = G(\mathbf{w}, z) = U_{L-1}\sigma(Z_{L-1}W_{L-1})W_L = Z_L W_L, \tag{3}$$

where $\sigma(\cdot)$ represents the action of ReLU operation, $Z_i^{d_i \times k_i} = U_{i-1}\sigma(Z_{i-1}W_{i-1})$, for $i = 2, \ldots L$, $z = \text{vec}(Z_1)$, $d_L = d$ and $W_L \in \mathbb{R}^{k_L \times k}$.

Fixing this architectural framework, we define the deep network prior as follows:

Definition 1. *A given image $x \in \mathbb{R}^{d \times k}$ is said to obey an untrained neural network prior if it belongs to a set $\mathcal{S}$ defined as: $\mathcal{S} := \{x | x = G(\boldsymbol{w}; z)\}$, where z is a (randomly chosen, fixed) latent code vector and $G(\boldsymbol{w}; z)$ has the form in Eq. 3.*

3 Solution methodology

The loss function in Eq.2 can be solved using gradient descent by optimizing over weights $\mathbf{w}$ and we call this formulation Net-GD, which has been described in Algorithm 1. The gradients $\nabla \mathcal{L}_2(\mathbf{w})$ in Step 3 of Algorithm 1 can be computed using back-propagation in standard frameworks for deep network optimization such as TensorFlow and PyTorch, with learning rate η. The weights $\mathbf{w}^0$ are initialized according to He initialization (scaled Gaussian random) [24].

A projected gradient descent based algorithm (Algorithm 2) was also tested out. Algorithm 2 consists of two parts; in Line 3 we compute the Wirtinger gradient [14] with learning rate η as follows:

$$v^t \leftarrow x^t - \eta A^\top (Ax^t - y \circ \text{sign}(Ax^t))$$

and execute one step for solving an unconstrained phase retrieval problem with gradient descent. In Lines 4 and 5, we estimate the weights of the deep network prior with noisy input v^t and ensure that the output $\mathbf{w}^t$ and subsequently the image estimate $x^t = G(\mathbf{w}^t; z)$ lies in the range of the decoder $G(\cdot)$ outlined by set $\mathcal{S}$, which is the main assumption of our problem formulation. The sub-problem in Step 4, which is a denoising problem is solved using gradient descent, which can be implemented via back-propagation in standard computational frameworks. An in-depth theoretical inspection of Algorithm 2 can be found in [25].

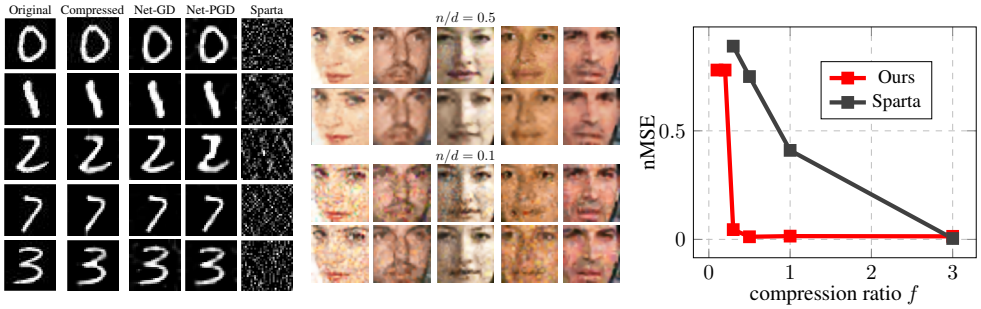
Initialization: Due to the non-convex nature of the problem, appropriate initialization is crucial. For Algorithm 2, we initialize $x^0 = G(\mathbf{w}^0; z)$, where $\mathbf{w}^0$ are assigned according to He initialization. In our experiments (Section 4) we note that this initialization is sufficient for both Net-GD and Net-PGD to succeed for CPR. In Section 4 we show experimental evidence to support this claim.

4 Experimental results

Dataset: We use images from the MNIST database (Figure 1(a)) and CelebA database (Figure 1(b)) to test our algorithms and reconstruct 6 grayscale (MNIST, 28×28 pixels ($d = 784$)) and

Algorithm 2 Net-PGD for compressive phase retrieval.

Input: $A, z = \text{vec}(Z_1), \eta, x^0 = G(\mathbf{w}^0; z)$.
while termination condition not met **do**
  $v^t \leftarrow x^t - \eta \nabla \mathcal{L}_1(x^t)$ {gradient step for phase retrieval}
  $\mathbf{w}^t \leftarrow \arg\min_{\mathbf{w}} \|v^t - G(\mathbf{w}; z)\|$ {projection to range of deep network}
  $x^{t+1} \leftarrow G(\mathbf{w}^t; z)$
end while
Output $\hat{x} \leftarrow x^T$.



(a) (c) (c)

Figure 1: (CPR) Reconstructed images from magnitude-only measurements (a) at compression rate of $n/d = 0.3$ for MNIST, (b) at compression rates of $n/d = 0.1, 0.5$ for CelebA with (row 1,3) Net-GD and (row 2,4) Net-PGD, (c) nMSE at different compression rates $f = n/d$ for MNIST.

5 RGB (CelebA) images. The CelebA dataset images are center cropped to size $64 \times 64 \times 3$ ($d = 12288$). The pixel values of all images are scaled to lie between 0 and 1. **Deep network architecture:** We first optimize the deep network architecture which fit our example images such that $x^* \approx G(\mathbf{w}^*; z)$ (referred as "compressed" image). For MNIST images, the architecture was fixed to a 2 layer configuration $k_1 = 15, k_2 = 15, k_3 = 10$, and for CelebA images, a 3 layer configuration with $k_1 = 120, k_2 = 15, k_3 = 15, k_4 = 10$. **Measurement setup:** We use a Gaussian measurement matrix of size $n \times d$ with n varied such that $n/d = 0.1, 0.2, 0.3, 0.5, 1, 3$. The elements of A are picked such that $A_{i,j} \sim \mathcal{N}(0, 1/n)$ and we report averaged reconstruction error values over 10 different instantiations of A for a fixed image (image of digit '0' from MNIST), network configuration and compression ratio n/d . **Performance metrics:** We compare reconstruction quality using normalized Mean-Squared Error (nMSE), which is calculated as $\|\hat{x} - x^*\|^2/\|x^*\|^2$ and plot its variation with different compression rates $f = n/d$ for all the algorithms tested averaged over all trials for MNIST in Figure 1 (c). **Algorithms and baselines:** We implement 3 schemes based on *untrained* priors for solving CPR , (i) Net-GD (ii) Net-PGD and finally (iii) Sparse Truncated Amplitude Flow (Sparta) [16] which is state-of-art, with sparse prior in DCT basis for both datasets. We note that both Net-GD and Net-PGD outperform Sparta (Figure 1(c)). **Implementation details:** Both Net-GD and Net-PGD were implemented using the PyTorch framework with Python 3 and using GPU support, using SGD optimizer. For implementing Sparta algorithm, the algorithm from [16] was implemented in MATLAB. We report the average running times for different algorithms across different measurement levels for examples from MNIST is 25.59s (Net-GD), 28.46s (Net-PGD), 3.80s (Sparta-DCT). **Goodness of random initialization:** We perform rigorous experiments to assert that He random initialization of the weights $\mathbf{w}^0$ of the neural network, ensure that the initial estimate $x^0 = G(\mathbf{w}^0; z)$ is good for Net-PGD. We denote the distance of initialization as $\delta_i = \|x^0 - \hat{x}\|/\|\hat{x}\|$ and report the values of δ_i for the trials in which $\|\hat{x} - x^*\|/\|x^*\| < 0.1$ in Table 1.

Table 1: Distance of initial estimate x^0 in Net-PGD (Algorithm 2)

n/d	d	channel configuration	nMSE of $\hat{x}$	average δ_i values
0.2	784 (MNIST)	15, 15, 10	0.098	0.914
0.5	784 (MNIST)	15, 15, 10	0.018	0.942
0.4	12288 (CelebA)	120, 15, 15,10	0.020	0.913
0.6	12288 (CelebA)	120, 15, 15,10	0.015	0.915

5 Acknowledgements

This work was supported in part by NSF grants CAREER CCF-2005804, CCF-1815101, and a faculty fellowship from the Black and Veatch Foundation.

References

[1] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[2] C. Dong, C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[3] J. Chang, C. Li, B. Póczos, and B. Kumar. One network to solve them all—solving linear inverse problems using deep projection models. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5889–5898. IEEE, 2017.

[4] C. Metzler, P. Schniter, A. Veeraraghavan, and R. Baraniuk. prdeep: Robust phase retrieval with a flexible deep network. In *International Conference on Machine Learning*, pages 3498–3507, 2018.

[5] A. Bora, A. Jalal, E. Price, and A. Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546. JMLR. org, 2017.

[6] P. Hand, O. Leong, and V. Voroninski. Phase retrieval under a generative prior. In *Advances in Neural Information Processing Systems*, pages 9136–9146, 2018.

[7] T. Lillicrap Y. Wu, M. Rosca. Deep compressed sensing. *arXiv preprint arXiv:1905.06723*, 2019.

[8] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[9] R. Heckel and P. Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. In *International Conference on Learning Representations*, 2018.

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

[11] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising with block-matching and 3d filtering. In *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, volume 6064, page 606414. International Society for Optics and Photonics, 2006.

[13] V. Papyan, Y. Romano, J. Sulam, and M. Elad. Convolutional dictionary learning via local processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5296–5304, 2017.

[14] Y. Chen and E. Candes. Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *Advances in Neural Information Processing Systems*, pages 739–747, 2015.

[15] G. Jagatap and C. Hegde. Fast, sample-efficient algorithms for structured phase retrieval. In *Advances in Neural Information Processing Systems*, pages 4917–4927, 2017.

[16] G. Wang, L. Zhang, G. Giannakis, M. Akçakaya, and J. Chen. Sparse phase retrieval via truncated amplitude flow. *IEEE Transactions on Signal Processing*, 66(2):479–491, 2017.

[17] E. Candes, T. Strohmer, and V. Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

[18] F. Shamshad, F. Abbas, and A. Ahmed. Deep ptych: Subsampled fourier ptychography using generative priors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7720–7724. IEEE, 2019.

[19] G. Jagatap and C. Hegde. Sample-efficient algorithms for recovering structured signals from magnitude-only measurements. *IEEE Transactions on Information Theory*, 2019.

[20] T. Cai, X. Li, and Z. Ma. Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. *The Annals of Statistics*, 44(5):2221–2251, 2016.

[21] R. Hyder, V. Shah, C. Hegde, and S. Asif. Alternating phase projected gradient descent with generative priors for solving compressive phase retrieval. *arXiv preprint arXiv:1903.02707*, 2019.

[22] V. Shah and C. Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4609–4613. IEEE, 2018.

[23] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[25] G. Jagatap and C. Hegde. Algorithmic guarantees for inverse imaging with untrained network priors. In *Advances in Neural Information Processing Systems*, 2019.