Inverse Imaging using Deep Untrained Neural Networks

NeurIPS 2019

Gauri Jagatap and Chinmay Hegde

Iowa State University

Imaging models

Imaging models

Given a Lj-dimensional image signal ⌞ and a sensing operator $\hat{a}(\ ) : \mathrm{R}^{Lj}\ !\ \mathrm{R}^{ϐ}$, measurements ⌞ take the form:

$$⌞ = \hat{a}(⌞\ )$$

Task: Recover ⌞ from measurements ⌞.

- Posed as an optimization problem:

$$\hat{⌞} = \arg\min_{⌞} \hat{I}(⌞) = \arg\min_{⌞} k⌞\quad \hat{a}(⌞)k_2^2$$

- Lj-dimensional image –> requires $ϐ = ì\,(Lj)$ measurements in conventional sensing systems for stable estimation (i.e. $\hat{⌞} = ⌞$).
- $\hat{a}$ can in general be ill-posed –> exact recovery $\hat{⌞} = ⌞$ is not guaranteed.

Examples of inverse problems: observation y

$y = x + \varepsilon$ $y = \Phi x$ $y = Px$

- Introduce a regularization that makes the problem more tractable.
- Degrees of freedom of natural images is typically lower than N.
- Constrain the search space to this lower-dimensional set S.

$$\hat{x} = \arg\min_{x \in S} \mathcal{L}(x) = \arg\min_{x \in S} \| y - \mathcal{A}(x) \|_2^2$$

Leveraging concise representations for regularization

- Natural images have lower dimensional structure –> this can be enforced as a prior for inverse imaging problems.

Prior S	Data?	Guarantees?
Sparsity (w or w/o structure, total variation)	No	Yes
Deep generative priors	Yes	Yes, Limited

Table 1: Low-dimensional priors

Leveraging concise representations for regularization

- Natural images have lower dimensional structure –> this can be enforced as a prior for inverse imaging problems.

Prior S	Data?	Guarantees?
Sparsity (w or w/o structure, total variation)	No	Yes
Deep generative priors	Yes	Yes, Limited

Table 1: Low-dimensional priors

! Are there other lower-dimensional representations that are more efficient?

Deep image prior

Prior S	Data?	Guarantees?
Sparsity (w or w/o structure, total variation)	No	Yes
Deep generative priors	Yes	Yes, Limited
Deep image prior (+this paper)	No	No ! Yes

Table 2: Low-dimensional priors

Deep image prior[1]: Using untrained neural networks as priors.

[1]D. Ulyanov et. al., Deep image prior, IEEE CVPR, 2018.

[2]G. Jagatap and C. Hegde, "Algorithmic Guarantees for Inverse Imaging with Untrained Network Priors," NeurIPS (2019).

Deep image prior

Prior S	Data?	Guarantees?
Sparsity (w or w/o structure, total variation)	No	Yes
Deep generative priors	Yes	Yes, Limited
Deep image prior (+this paper)	No	No ! Yes

Table 2: Low-dimensional priors

Deep image prior[1]: Using untrained neural networks as priors.

Our contributions[2]:

- New applications of deep image prior for inverse imaging.
    - Linear compressive sensing.
    - Compressive phase retrieval.
- Algorithmic guarantees for reconstruction.

[1]D. Ulyanov et. al., Deep image prior, IEEE CVPR, 2018.
[2]G. Jagatap and C. Hegde, "Algorithmic Guarantees for Inverse Imaging with Untrained Network Priors," NeurIPS (2019).

Deep Neural Networks for Inverse Imaging

Trained deep neural networks for inverse imaging

- Deep neural networks have been used successfully for learning image representations.
- Autoencoders, generative adversarial networks, trained on thousands of images learn latent representations which are lower-dimensional.
- Exploit global statistics across dataset images.

Trained deep neural networks for inverse imaging

- Deep neural networks have been used successfully for learning image representations.
- Autoencoders, generative adversarial networks, trained on thousands of images learn latent representations which are lower-dimensional.
- Exploit global statistics across dataset images.

! Does the structure of the neural network impose a good prior for image-related problems?

Trained deep neural networks for inverse imaging

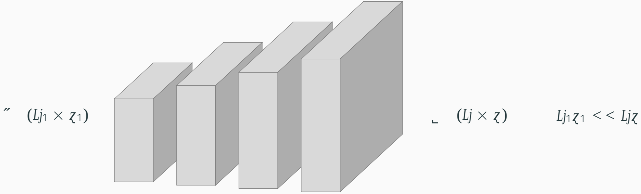
- Deep neural networks have been used successfully for learning image representations.
- Autoencoders, generative adversarial networks, trained on thousands of images learn latent representations which are lower-dimensional.
- Exploit global statistics across dataset images.

! Does the structure of the neural network impose a good prior for image-related problems?

! Can a neural network be used to represent one image, instead of thousands of images?

Untrained networks as priors

A given image ˛ 2 R$^{Lj \quad ζ}$ is said to obey an untrained neural network prior if it belongs to a set S defined as: S := f˛j˛ = ~(w; ˝)g where ˝ is a (randomly chosen, fixed, dimensionally smaller than ˛) latent code vector and ~(w; ˝) has the form as below.



˛ = ~(w; ˝) = l·$_{\hat{I} \quad 1}$ ($\hat{W}_{\hat{I} \quad 1}\check{S}_{\hat{I} \quad 1}$)$\check{S}_{\hat{I}}$ = $\hat{W}_{\hat{I}}\check{S}_{\hat{I}}$; (Heckel et. al. 2019)

( ) represents ReLU, $\hat{W}_{ζ}^{Ljζ \quad ζζ}$ = l·$_{ζ \quad 1}$ ($\hat{W}_{ζ \quad 1}\check{S}_{ζ \quad 1}$), for $ζ$ = 2:::$\hat{I}$, l· is bi-linear upsampling, ˝ = vec($\hat{W}_1$) 2 R$^{Lj_1 \quad ζ_1}$, $Lj_{\hat{I}}$ = Lj and $\check{S}_{\hat{I}}$ 2 R$^{ζ\hat{I} \quad ζ}$.

Applications of Untrained Neural Priors

Applications of Untrained Neural Priors in Inverse Imaging

Denoising

Super-resolution [3]

Inpainting [4]

[3] D. Ulyanov et. al., Deep image prior, IEEE CVPR, 2018.

[4] R. Heckel et. al., Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks, ICLR, 2019

Application to Compressive Imaging (Our contribution)

We consider two models for sIʒ 9 BdÙ r r ζ Ù imaging, with operator $\hat{a}(\ )$, such that $\llcorner = \hat{a}(\llcorner\ )$, and $\hat{a}$ takes the forms as below:

- Linear compressive sensing: $\llcorner = \ \llcorner$
- Compressive phase retrieval: $\llcorner = j\ \llcorner\ j$

where $\llcorner\ \ 2\ \mathrm{R}^{Lj}$, $\llcorner\ 2\ \mathrm{R}^{ɞ}$, and entries of are from $N(0; 1{=}ɞ)$ with $ɞ < Lj$.

Application to Compressive Imaging (Our contribution)

We consider two models for ꜱꞁꞫ ꞔ ʙꞇÙ ʳ ʳꞁ¸ Ù imaging, with operator $\hat{a}(\ )$, such that ⌞ = $\hat{a}$(⌞), and $\hat{a}$ takes the forms as below:

- Linear compressive sensing: ⌞ = ⌞
- Compressive phase retrieval: ⌞ = j ⌞ j

where ⌞ 2 R^{Lj}, ⌞ 2 $\mathsf{R}^{ɞ}$, and entries of are from $N(0;1{=}ɞ)$ with $ɞ < Lj$.

$!$ both of these problems are ill-posed in this form and require prior information (or regularization) to yield unique solutions.

Pose as the following optimization problem:

$$\min_{⌞;\mathsf{w}} k_{⌞} \quad \hat{a}(⌞)k_2 \quad \text{s.t.} \quad ⌞ = \sim(\mathsf{w};\text{˝}) \; 2 \; S$$

where weights w need to be estimated and S is the range of signals that can be represented as ⌞ = $\sim(\mathsf{w};\text{˝})$.

Projected gradient descent for compressive imaging with untrained neural priors

PGD for compressive sensing with untrained neural priors

Solve: $\min_{\llcorner 2 S} \hat{I}(\llcorner) = \min_{\llcorner 2 S} k_{\leftharpoondown} \quad \llcorner k^2$

Algorithm 1 Net-PGD for linear compressive sensing.

Input: $\leftharpoondown; \; ; ''; \; ; \imath = \log \frac{1}{}; \llcorner^0 = \sim(\mathsf{w}^0; '')$

for $? = 1; \quad ; \imath$ **do**

$\quad {}_{\text{,}}^{?} \quad \llcorner^{?} \quad {}^{>}(\; \llcorner^{?} \quad \leftharpoondown)$ {gradient step for least squares}

$\quad \mathsf{w}^{?} \quad \arg\min_{\mathsf{w}} k_{\text{,}}^{?} \quad \sim(\mathsf{w}; '')k$ {projection to S}

$\quad \llcorner^{?+1} \quad \sim(\mathsf{w}^{?}; '')$

end for

Output $\hat{\llcorner} \quad \llcorner^{\imath}$.

PGD for compressive phase retrieval with untrained priors

Solve: $\min_{x \in S} \hat{L}(x) = \min_{x \in S} \| y - |Ax| \|^2$

Algorithm 2 Net-PGD for compressive phase retrieval.

1: **Input:** $y, A, z = \text{vec}(\hat{W}_1), \eta, T = \log \frac{1}{\epsilon}, x^0$ s.t. $\|x^0 - x^*\| \le \delta_i \|x^*\|$.
2: **for** $t = 1, \cdots, T$ **do**
3: $p^t \leftarrow \text{sign}(Ax^t)$ {phase estimation}
4: $v^t \leftarrow x^t - \eta A^\top (Ax^t - y \circ p^t)$ {gradient step for phase retrieval}
5: $w^t \leftarrow \arg\min_{w} \|v^t - G(w; z)\|$ {projection to S}
6: $x^{t+1} \leftarrow G(w^t; z)$
7: **end for**
8: **Output** $\hat{x} \leftarrow x^T$.

Theoretical guarantees

Unique recovery : Set-restricted Isometry Property

To establish unique recovery of ⌞ from ⌞, we need the measurement matrix to satisfy a set-restricted isometry property as follows:

Lemma: Set-RIP for Gaussian matrices

If an image ⌞ 2 R^{Lj} has a decoder prior (captured in set S), where the decoder consists of weights **w** and piece-wise linear activation (ReLU), a random Gaussian matrix 2 $\mathrm{R}^{ɞ \; Lj}$ with elements from $N(0; 1{=}ɞ)$, satisfies $(S; 1 \quad ; 1 + \quad)$-RIP, with probability 1 Ủ ʂ ^{2}ɞ, as long as ɞ = ì ζ1/2 $\hat{\mathbb{P}}_{Ɇ=2}$ ζɆ log Lj , for small constant ʂ and 0 < < 1.

$$(1 \quad)k⌞k^2 \quad k \quad ⌞k^2 \quad (1 + \quad)k⌞k^2:$$

Proof sketch

- For a fixed linearized subspace, the image x has a representation of the form

$$x = U \hat{W} w,$$

where U absorbs all upsampling operations, $\hat{W}$ is latent code which is fixed and known and w is the direct product of all weight matrices with $w \in \mathbb{R}^{k_1}$.

- An oblivious subspace embedding (OSE) of x takes the form

$$(1 - \epsilon)\|x\|^2 \leq \|Sx\|^2 \leq (1 + \epsilon)\|x\|^2,$$

where S is a Gaussian matrix, and holds for all k_1-dimensional vectors w, with high probability as long as $m = O(k_1/\epsilon^2)$.

- Counting argument for the number of such linearized networks followed by union of subspaces argument to capture the range of a deep untrained network.

Convergence of Net-PGD for Linear Compressive Sensing

Suppose the sampling matrix satisfies $(S, 1 \quad, 1 + \quad)$-RIP with high probability then, Algorithm 1 produces $\hat{}$ such that $\|\hat{} \quad - \quad \|$ and requires $\log \frac{1}{}$ iterations.

Proof approach:

- $(S, 1 \quad, 1 + \quad)$-RIP for $\quad, {}^{t}, {}^{t+1} \in S$
- gradient update rule
- exact projection criterion $\|{}^{t+1} \quad {}^{t}\| \quad \| \quad {}^{t}\|$

to establish the contraction $\|{}^{t+1} \quad \| \quad \|{}^{t} \quad \|$, with < 1 to guarantee [illegible] for compressed sensing recovery.

Theoretical guarantees

Convergence of Net-PGD for Compressive Phase Retrieval

Suppose the sampling matrix $ɞ \; Ǉj$ satisfies $(S; 1 \quad ; 1 + \quad)$-RIP with high probability, Algorithm 2 produces $\hat{⌴}$, such that $k\hat{⌴} \quad ⌴ \; k$,
as long as the weights are initialized such that $k⌴^0 \quad ⌴ \; k \quad ʒ k⌴ \; k$
and the number of measurements is $ɞ = ì \quad ʒ_1 \prod_{ᵻ=2} ʒ_ᵻ \log Ǉj$.

- $(S; 1 \quad ; 1 + \quad)$-RIP for $⌴ \; ; ⌴^ʔ ; ⌴^{ʔ+1} \; 2 \; S$
- gradient update rule
- exact projection criterion $k⌴^{ʔ+1} \quad ¸^ʔ k \quad k⌴ \quad ¸^ʔ k$
- bound on the ʙşƐʳǛ Ǜʳʔʒ ɘ Ɛʔʒʓɞ error $k''^ʔ_ʙ k_2$,
  $''^ʔ_ʙ :=\; ^> \; ⌴ \quad (1 \quad \text{sign}(\; ⌴ \;) \quad \text{sign}(\; ⌴^ʔ))$ (requires good initialization)

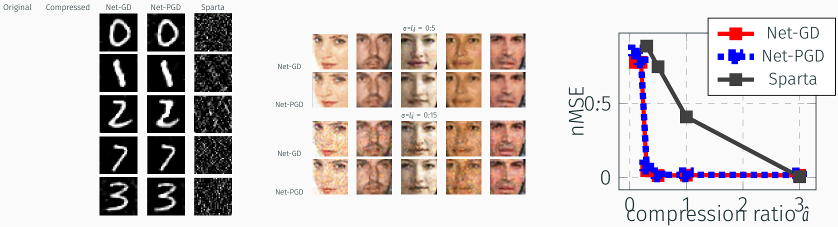
to establish the contraction $k⌴^{ʔ+1} \quad ⌴ \; k \quad k⌴^ʔ \quad ⌴ \; k$, with $\quad < 1$ to guarantee ȽʓşƐȽ Ƚʒɞ ǛƐȡ şʓɞ¸ ǛȡÊǛɞşǛ ʓâ ÞǛʔ◭Ė~P for compressive phase retrieval.

Experiments

Linear compressive sensing

Figure 1: (CS) Reconstructed images from linear measurements (at compression rate $m/n = 0.1$) with (a) $m = 78$ measurements for examples from MNIST, (b) $m = 1228$ measurements for examples from CelebA, and (c) nMSE at different compression rates $\alpha = m/n$ for MNIST.

Net-GD: Solve $\min_{w} \| y - A(G(w; z)) \|_2^2$



(a) (b) (c)

Figure 2: (CPR) Reconstructed images from magnitude-only measurements (a) at compression rate of ϵ=Lj = 0:3 for MNIST, (b) for CelebA with Net-GD and Net-PGD, (c) nMSE at different compression rates $\hat{a}$ = ϵ=Lj for MNIST.

Net-GD: Solve $\min_{w} k_{\leftarrow} \quad \hat{a}(\sim(w;\text{˝}))k_2^2$

Conclusion and future directions

- Our contributions:
  - Novel applications of untrained neural priors to two problems: compressive sensing and phase retrieval with superior empirical performance.
  - Algorithmic guarantees for convergence of PGD for both applications.
- Future directions:
  - Explore other applications such as signal demixing, modulo imaging.
  - Theoretical guarantees for projection oracle.
  - Testing invertible architectures like Glow, instead of decoder structure as prior.

Thank you!

Deep network configuration

- Fit our example images such that ⌞ ~(w ;˝) (referred as "compressed" image).
- For MNIST images, the architecture was fixed to a 2 layer configuration $\zeta_1 = 15; \zeta_2 = 15; \zeta_3 = 10$.
- For CelebA images, a 3 layer configuration with $\zeta_1 = 120; \zeta_2 = 15; \zeta_3 = 15; \zeta_4 = 10$ was sufficient to represent most images.
- Both architectures use bilinear upsampling operators each with upsampling factor of 2, $l\cdot_{\not{E}}^{"2}; \not{E} = f1; 2; 3g$.
- The outputs after each ReLU operation are normalized, by calling for batch normalization subroutine in Pytorch.
- Finally a sigmoid activation is added to the output of the deep network, which smoothens the output; however this is not mandatory for the deep network configuration to work.