Inverse Imaging using Deep Untrained Neural Networks

NeurIPS 2019

Gauri Jagatap and Chinmay Hegde

Iowa State University
Imaging models
Given a $d$-dimensional image signal $x^*$ and a sensing operator $f(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^n$, measurements $y$ take the form:

$$y = f(x^*)$$

**Task:** Recover $x^*$ from measurements $y$.

- Posed as an optimization problem:
  $$\hat{x} = \arg\min_x L(x) = \arg\min_x \|y - f(x)\|_2^2$$

- $d$-dimensional image $\rightarrow$ requires $n = O(d)$ measurements in conventional sensing systems for stable estimation (i.e. $\hat{x} = x^*$).

- $f$ can in general be ill-posed $\rightarrow$ exact recovery $\hat{x} = x^*$ is not guaranteed.
Examples of inverse problems: observation $y$

- $y = x + \varepsilon$
- $y = Mx$
- $y = Dx$

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

$$\hat{x} = \arg\min_{x \in S} L(x) = \arg\min_{x \in S} \|y - f(x)\|_2^2$$
Leveraging concise representations for regularization

- Natural images have lower dimensional structure \(\rightarrow\) this can be enforced as a prior for inverse imaging problems.

<table>
<thead>
<tr>
<th>Prior (S)</th>
<th>Data?</th>
<th>Guarantees?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparsity (w or w/o structure, total variation)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Deep generative priors</td>
<td>Yes</td>
<td>Yes, Limited</td>
</tr>
</tbody>
</table>

Table 1: Low-dimensional priors
• Natural images have lower dimensional structure \(\rightarrow\) this can be enforced as a prior for inverse imaging problems.

<table>
<thead>
<tr>
<th>Prior (S)</th>
<th>Data?</th>
<th>Guarantees?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparsity (w or w/o structure, total variation)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Deep generative priors</td>
<td>Yes</td>
<td>Yes, Limited</td>
</tr>
</tbody>
</table>

Table 1: Low-dimensional priors

\(\rightarrow\) Are there other lower-dimensional representations that are more efficient?
Deep image prior

<table>
<thead>
<tr>
<th>Prior $S$</th>
<th>Data?</th>
<th>Guarantees?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparsity (w or w/o structure, total variation)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Deep generative priors</td>
<td>Yes</td>
<td>Yes, Limited</td>
</tr>
<tr>
<td>Deep image prior (+this paper)</td>
<td>No</td>
<td>No $\rightarrow$ Yes</td>
</tr>
</tbody>
</table>

**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.

Deep image prior

<table>
<thead>
<tr>
<th>Prior $S$</th>
<th>Data?</th>
<th>Guarantees?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparsity (w or w/o structure, total variation)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Deep generative priors</td>
<td>Yes</td>
<td>Yes, Limited</td>
</tr>
<tr>
<td>Deep image prior (+this paper)</td>
<td>No</td>
<td>No $\rightarrow$ Yes</td>
</tr>
</tbody>
</table>

**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.

---


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?
Deep Image Prior : Untrained Neural Priors

Untrained networks as priors

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 \( S \) defined as: \( S := \{ x | x = G(w; z) \} \) where 

\[
S := \{ x | x = G(w; z) \}
\]

\( z \) is a (randomly chosen, fixed, dimensionally smaller than \( x \)) latent code vector and \( G(w; z) \) has the form as below.

\[
x = G(w, z) = U_{L-1} \sigma(Z_{L-1}W_{L-1})W_L = Z_LW_L, \text{(Heckel et. al. 2019)}
\]

\( \sigma(\cdot) \) represents ReLU, \( Z_i^{d_i \times k_i} = U_{i-1} \sigma(Z_{i-1}W_{i-1}) \), for \( i = 2...L, U \) is bi-linear upsampling, \( z = \text{vec}(Z_1) \in \mathbb{R}^{d_1 \times k_1} \), \( d_L = d \) and \( W_L \in \mathbb{R}^{k_L \times k} \).
Applications of Untrained Neural Priors
Applications of Untrained Neural Priors in Inverse Imaging

Denoising

Super-resolution

Inpainting

4. R. Heckel et. al., Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks, ICLR, 2019
We consider two models for compressive imaging, with operator $f(\cdot)$, such that $y = f(x^*)$, and $f$ takes the forms as below:

- Linear compressive sensing: $y = Ax^*$
- Compressive phase retrieval: $y = |Ax^*|$

where $x^* \in \mathbb{R}^d$, $y \in \mathbb{R}^n$, and entries of $A$ are from $\mathcal{N}(0, 1/n)$ with $n < d$. 
We consider two models for compressive imaging, with operator $f(\cdot)$, such that $y = f(x^*)$, and $f$ takes the forms as below:

- Linear compressive sensing: $y = Ax^*$
- Compressive phase retrieval: $y = |Ax^*|

where $x^* \in \mathbb{R}^d$, $y \in \mathbb{R}^n$, and entries of $A$ are from $\mathcal{N}(0, 1/n)$ with $n < d$.

→ 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:

$$\min_{x,w} \|y - f(x)\|_2 \quad \text{s.t.} \quad x = G(w, z) \in S$$

where weights $w$ need to be estimated and $S$ is the range of signals that can be represented as $x = G(w, z)$. 
Projected gradient descent for compressive imaging with untrained neural priors
Solve: $\min_{x \in S} L(x) = \min_{x \in S} \|y - Ax\|^2$

Algorithm 1 Net-PGD for linear compressive sensing.

1: Input: $y, A, z, \eta, T = \log \frac{1}{\epsilon}, x^0 = G(w^0; z)$
2: for $t = 1, \cdots, T$ do
3: \hspace{1em} $v^t \leftarrow x^t - \eta A^\top (Ax^t - y)$ \quad \{gradient step for least squares\}
4: \hspace{1em} $w^t \leftarrow \arg \min_w \|v^t - G(w; z)\|$ \quad \{projection to $S$\}
5: \hspace{1em} $x^{t+1} \leftarrow G(w^t; z)$
6: end for
7: Output $\hat{x} \leftarrow x^T$. 
Solve: $\min_{x \in S} L(x) = \min_{x \in S} \|y - |Ax|\|^2$

**Algorithm 2** Net-PGD for compressive phase retrieval.

1. **Input:** $A, z = \text{vec}(Z_1), \eta, T = \log \frac{1}{\epsilon}, x^0$ s.t. $\|x^0 - x^*\| \leq \delta_i\|x^*\|$.
2. **for** $t = 1, \cdots, T$ **do**
   3. $p^t \leftarrow \text{sign}(Ax^t)$ \hspace{2cm} \{phase estimation\}
   4. $v^t \leftarrow x^t - \eta A^T (Ax^t - y \circ p^t)$ \hspace{2cm} \{gradient step for phase retrieval\}
   5. $w^t \leftarrow \arg \min_w \|v^t - G(w; z)\|$ \hspace{2cm} \{projection to $S$\}
   6. $x^{t+1} \leftarrow G(w^t; z)$
3. **end for**
4. **Output** $\hat{x} \leftarrow x^T$. 
Theoretical guarantees
To establish unique recovery of $x^*$ from $y$, we need the measurement matrix $A$ to satisfy a set-restricted isometry property as follows:

**Lemma: Set-RIP for Gaussian matrices**

If an image $x \in \mathbb{R}^d$ has a decoder prior (captured in set $\mathcal{S}$), where the decoder consists of weights $w$ and piece-wise linear activation (ReLU), a random Gaussian matrix $A \in \mathbb{R}^{n \times d}$ with elements from $\mathcal{N}(0, 1/n)$, satisfies $(\mathcal{S}, 1 - \alpha, 1 + \alpha)$-RIP, with probability $1 - e^{-c\alpha^2 n}$, as long as $n = O\left(\frac{k_1}{\alpha^2} \sum_{l=2}^{L} k_l \log d\right)$, for small constant $c$ and $0 < \alpha < 1$.

$$(1 - \alpha)\|x\|^2 \leq \|Ax\|^2 \leq (1 + \alpha)\|x\|^2.$$
Proof sketch

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

$$x = UZw,$$

where $U$ absorbs all upsampling operations, $Z$ 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

$$(1 - \alpha)\|x\|^2 \leq \|Ax\|^2 \leq (1 + \alpha)\|x\|^2,$$

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

- Counting argument for the number of such linearized networks followed by union of subspaces argument to capture the range of a deep untrained network.
Theoretical guarantees: Convergence of Net-PGD

Convergence of Net-PGD for Linear Compressive Sensing

Suppose the sampling matrix $A^{n \times d}$ satisfies $(S, 1 - \alpha, 1 + \alpha)$-RIP with high probability then, Algorithm 1 produces $\hat{x}$ such that $||\hat{x} - x^*|| \leq \epsilon$ and requires $T \propto \log \frac{1}{\epsilon}$ iterations.

Proof approach:

- $(S, 1 - \alpha, 1 + \alpha)$-RIP for $x^*, x^t, x^{t+1} \in S$
- gradient update rule
- exact projection criterion $||x^{t+1} - v^t|| \leq ||x^* - v^t||$

to establish the contraction $||x^{t+1} - x^*|| \leq \nu ||x^t - x^*||$, with $\nu < 1$ to guarantee linear convergence of Net-PGD for compressed sensing recovery.
### Theoretical guarantees

#### Convergence of Net-PGD for Compressive Phase Retrieval

Suppose the sampling matrix \( A^{n \times d} \) satisfies \((\mathcal{S}, 1 - \alpha, 1 + \alpha)\)-RIP with high probability, Algorithm 2 produces \( \hat{x} \), such that \( \|\hat{x} - x^*\| \leq \epsilon \), as long as the weights are initialized such that \( \|x^0 - x^*\| \leq \delta_i \|x^*\| \) and the number of measurements is \( n = O \left( k_1 \sum_{l=2}^{L} k_l \log d \right) \).

- \((\mathcal{S}, 1 - \alpha, 1 + \alpha)\)-RIP for \( x^*, x^t, x^{t+1} \in \mathcal{S} \)
- gradient update rule
- exact projection criterion \( \|x^{t+1} - v^t\| \leq \|x^* - v^t\| \)
- bound on the phase estimation error \( \|\varepsilon_p^t\|_2 \),
  \[ \varepsilon_p^t := A^\top A x^* \circ (1 - \text{sign}(A x^*) \circ \text{sign}(A x^t)) \] (requires good initialization)

To establish the contraction \( \|x^{t+1} - x^*\| \leq \nu \|x^t - x^*\| \), with \( \nu < 1 \) to guarantee **local linear convergence of Net-PGD** for compressive phase retrieval.
Experiments
Figure 1: (CS) Reconstructed images from linear measurements (at compression rate $n/d = 0.1$) with (a) $n = 78$ measurements for examples from MNIST, (b) $n = 1228$ measurements for examples from CelebA, and (c) nMSE at different compression rates $f = n/d$ for MNIST.

Net-GD: Solve $\min_w \| y - f(G(w; z)) \|_2^2$
Compressive phase retrieval

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

Net-GD: Solve $\min_w \| y - f(G(w; z)) \|_2^2$
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 $x^* \approx G(w^*; z)$ (referred as “compressed” image).

• For MNIST images, the architecture was fixed to a 2 layer configuration $k_1 = 15, k_2 = 15, k_3 = 10$.

• For CelebA images, a 3 layer configuration with $k_1 = 120, k_2 = 15, k_3 = 15, k_4 = 10$ was sufficient to represent most images.

• Both architectures use bilinear upsampling operators each with upsampling factor of 2, $U_l^{\uparrow 2}, l = \{1, 2, 3\}$.

• 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 smoothen the output; however this is not mandatory for the deep network configuration to work.