Faster Guaranteed GAN-based Recovery in Linear Inverse Problems

Yoram Bresler
Coordinated Science Lab and Dept. of ECE
University of Illinois, Urbana-Champaign

Work with

• Ankit Raj

• Yuqi Li

Acknowledgements: NSF IIS 14-47879 and Sandia National Labs

IMA
October 15, 2019
The Image Recovery Problem

\[ y = Ax + v, A \in \mathbb{R}^{m \times n}, v \sim \mathcal{N}(0, \sigma^2 I) \]

- Ill-posed problem
- Underdetermined problem: fat measurement matrix \( A \)
- Need regularizer/prior to find a unique solution
Five Waves of Image Reconstruction

1. Analytic recon (FBP, Fourier)
2. MBIR (MRF, Bayesian)
3. Compressed Sensing
4. Shallow Learning
5. Deep Learning

Modified from JC Ye's ISBI'2018 presentation
Great hopes ...

Image Reconstruction Is a New Frontier of Machine Learning
Much excitement ...

Science Highlights:  June 10, 2019

Artificial intelligence enables low-dose CT scans, faster scan time

Radiologists give high rating to low-dose CT processed with a novel deep-learning technique
Deep Learning for Inverse Problems

Competitive performance of a modularized deep neural network compared to commercial algorithms for low-dose CT image reconstruction

Hongming Shan¹, Atul Padole², Fatemeh Homayounieh², Uwe Kruger², Ruhani Doda Khera², Chayanin Nitiwarangkul²,³, Mannudeep K. Kalra²* and Ge Wang¹*
Deep Learning for Inverse Problems

**Fig. 2 | Comparison between the best DL reconstruction and the best IR reconstruction for abdomen and chest regions across three major vendors (A, B and C) and three readers (R1, R2 and R3).**

- **a-f**, Histograms showing the number of cases per class in 20 cases each for the abdomen from vendor A (a), the chest from vendor A (b), the abdomen from vendor B (c), the chest from vendor B (d), the abdomen from vendor C (e) and the chest from vendor C (f) respectively. The text in the grey box above each plot gives the significant results evaluated by the sign test at 5% significant level for each reader.
Challenges for the Imaging Community in DL for IPs

- A principled way to derive good architectures?
- Use of domain expertise?
- Link to existing inverse problem and signal processing wisdom?
- Rigorous analysis?
- Theoretical guarantees?
Structures
Learned ISTA (LISTA)

\[ \hat{x}^{k+1} = S_{\lambda/L} \left( A_e y + W \hat{x}^k \right) \]

- Time unroll the iteration for K steps

- Learn the \( A_e \) and \( W \) matrices (minimize squared loss) with "backprop-through-time"

- Get the best approximate solution within K iterations

Gregor et al, ICML, 2010
Learned Projected Gradient

Algorithm 1 Relaxed projected gradient descent (RPGD)

**Input:** \( \mathbf{H}, \mathbf{y}, \mathbf{A} \), nonlinear operator \( F \), step size \( \gamma > 0 \), positive sequence \( \{c_n\}_{n \geq 1} \), \( \mathbf{x}_0 = \mathbf{Ay} \in \mathbb{R}^N \), \( \alpha_0 \in (0, 1] \).

**Output:** reconstructions \( \{\mathbf{x}_k\} \), relaxation parameters \( \{\alpha_k\} \).

\[
\begin{align*}
k & \leftarrow 0 \\
\text{while} & \text{ not converged do} \\
\quad & \mathbf{z}_k = F(\mathbf{x}_k - \gamma \mathbf{H}^T \mathbf{H} \mathbf{x}_k + \gamma \mathbf{H}^T \mathbf{y}) \\
\quad & \text{if } k \geq 1 \text{ then} \\
\quad & \quad \text{if } \|\mathbf{z}_k - \mathbf{x}_k\|_2 > c_k \|\mathbf{z}_{k-1} - \mathbf{x}_{k-1}\|_2 \text{ then} \\
\quad & \quad \quad \alpha_k = c_k \|\mathbf{z}_{k-1} - \mathbf{x}_{k-1}\|_2 / \|\mathbf{z}_k - \mathbf{x}_k\|_2 \alpha_{k-1} \\
\quad & \quad \text{else} \\
\quad & \quad \quad \alpha_k = \alpha_{k-1} \\
\quad & \text{end if} \\
\quad & \text{end if} \\
\quad & \mathbf{x}_{k+1} = (1 - \alpha_k) \mathbf{x}_k + \alpha_k \mathbf{z}_k \\
\quad & k \leftarrow k + 1 \\
\text{end while}
\]

Gupta et al, TMI, 2018

Replacing the projection operator with CNN

Courtesy of JC Ye
Desirata

- Decouple the learning of an image prior from the forward problem
- Fast operation (in the inference stage)
- Theoretical analysis/guarantees
Neural-Network Based Priors for IP

- **Plug-and-play Priors for Image Reconstruction**
  - Image prior expressed by a denoiser in the ADMM set-up
  - Not guaranteed to learn the distribution (prior) of the images

- **One Network to Solve Them All**
  - Train end-to-end proximal map onto the data set in an adversarial way
  - Training - highly sensitive to parameters
  - Set of images for the proximal map is not well-defined
  - No theoretical guarantee on convergence

Generative Models as a Prior?

- Successful in modeling data distribution \( P_x \)
- Learn to generate samples from \( P_x \)
- State-of-the-art: Generative Adversarial Networks (GANs)
  - Traditional priors replaced by the GAN-based prior
  - Recovery with \( \ll \) measurements than sparsity-based methods
- Typical generative model:
  \[
  G: z \in \mathbb{R}^k \mapsto x \in \mathbb{R}^n \quad n \ll k
  \]
- Problem formulation:
  \[
  \hat{x} = \text{argmin}_{x \in \text{Range}(G)} \| y - Ax \|_2^2
  \]
Generative Adversarial Networks (GoodFellow, NIPS, 2016)

Figure adopted from
• BEGAN: Boundary Equilibrium Generative Adversarial Networks, 17’03
• StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, ‘16.12
Related Works - CSGM

Compressed Sensing using Generative Models (Bora et al. 2017)

- Solve for the latent-space, \( z \), using iterative algorithm

\[
\hat{x} = G(\hat{z}), \hat{z} = \arg\min_{z} \| y - AG(z) \|^2
\]

- Generative models: VAEs for MNIST and GANs for CelebA
- Requires multiple initializations and chooses the best
- Computation-heavy

Related Works - PGDGAN

- Solve directly for the signal $x$
  \[
  \min_x f = \|y - Ax\|^2 \text{ s.t. } x = G(z)
  \]
- Projected Gradient Descent (PGD)
- Idea: project gradient update onto generative model manifold
- Inner+outer iterations
- Computation heavy: computes network Jacobian every inner iteration

\[
  x_k \rightarrow I - \eta_1 \nabla_x f \rightarrow w_k \rightarrow g(z) = \left\| G(z) - w_k \right\|^2 \rightarrow \hat{z}_{k+1} \leftarrow \arg\min_z g(z) \rightarrow G \rightarrow x_{k+1}
\]

Shah, V., & Hegde, C. Solving linear inverse problems using GAN priors: An algorithm with provable guarantees. ICASSP 2018
Issues with These Methods

- **CSGM**: no convergence guarantee
  - Requires several random initializations in $z$
  - Computation of Jacobian of the network: $T_{outer} \times \#\text{restarts}$

- **PGDGAN**: non-convex optimization problem in the inner-loop
  - Computation of Jacobian of the network: $T_{outer} \times T_{inner}$
  - Extremely slow because of inner-loop
  - Conditions required for convergence stringent and cannot be achieved with a moderate number of measurements
Proposed Method: NPGD

- Learn a network-based projector onto R(G)
  - Eliminates inner-loop
  - Eliminates expensive computation of Jacobian of G
  - Recovery guarantee if A satisfies conditions.
  - Learned A provides recovery with fewer measurements than random A
  - NPGD solves different linear inverse problems, e.g., compressed sensing, inpainting, super-resolution

\[
x_k \xrightarrow{I - \eta \nabla_x f} W_k \xrightarrow{P_G = G \circ G^\dagger} x_{k+1}
\]
Network-Based Projector

- **Goal:** learn a projector onto the set of natural images (onto $\text{Range}(G)$)
- **Assumption:** the trained generator $G^*$ defines this set
- $G^\dagger_\theta$ approximates a non-linear least squares pseudo-inverse of $G^*$
- $G^*(z)$ is perturbed by noise to train on points outside $R(G^*)$
- $P_G = G \circ G^\dagger_\theta$ is trained to project outside points onto $R(G^*)$
- $G^*$ remains fixed while training the projector

\[
\mathcal{L}(\theta) = \mathbb{E}_z \left[ \left\| G^* \left( G^\dagger_\theta(G^*(z) + \nu) \right) - G^*(z) \right\|^2 \right] \\
+ \mathbb{E}_z \left[ \lambda \left\| G^\dagger_\theta(G^*(z) + \nu) - z \right\|^2 \right]
\]
Network-based Projected Gradient Descent (NPGD)

- Maximum likelihood estimate of $x$ using GAN prior:
  \[
  \hat{x}_{MLE} = \arg \min_{x \in R(G)} \|y - Ax\|_2^2
  \]
  - Solve via PGD with a trained projector

---

**Algorithm 1** Network-based Projected Gradient Descent

**Input:** loss function $f$, $A$, $y$, $G$, $G^\dagger$

**Parameter:** step size $\eta (= \frac{1}{\beta})$

**Output:** an estimate $\hat{x} \in R(G)$

1: Let $t = 0$, $x_0 = A^T y$.
2: while $t < T$ do
3: \[ w_t := x_t - \eta A^T (Ax_t - y) \]
4: \[ x_{t+1} := G(G^\dagger(w_t)) \]
5: end while
6: return $\hat{x} = x_T$
Network Architectures

- **MNIST**: 28x28 grayscale images
  - Generator and discriminator adapted from DCGAN.
  - $G$: 4 transposed convolution layers, $D$: 4 convolution layers,
  - $G^{\dagger}_{\theta}$ similar to $D$ except last layer produces latent vector

- **CelebA**: 64x64 color images of celebrities’ faces
  - DCGAN’s 5 layer structure

- **LSUN-Church**: 64x64 color image of church outdoor architecture
  - Self-attention GAN (SAGAN) - $G$: 4 transpose convolution layers, $D$: 4 conv layers
    - with the self-attention mechanism in 3rd and 4th layers
  - Attention mechanism learns which part of the image to be emphasized
  - Better for capturing high-resolution details using information from different feature locations
Training Details

- Images scaled between [-1, 1]
- Latent variable is $z \sim \mathcal{N}(0, I_k)$
- Vanilla GAN loss for the DCGAN network
- Wasserstein loss with spectral normalization for the SAGAN
- Adam optimizer
- Mini-batch size: 128 for MNIST and 64 for CelebA and LSUN
Convergence Analysis

Restricted eigenvalue condition (REC):
Let $S \subset \mathbb{R}^n$. Matrix $A \in \mathbb{R}^{m \times n}$ satisfies the restricted eigenvalue condition $REC(S, \alpha, \beta)$ if the following holds for all $x_1, x_2 \in S$:

$$\alpha \|x_1 - x_2\|^2 \leq \|A(x_1 - x_2)\|^2 \leq \beta \|x_1 - x_2\|^2$$

If $\beta / \alpha \approx 1$ then matrix $A$ is a near-isometry on $S$. 
Convergence Analysis

Restricted eigenvalue condition (REC):
Let \( S \subset \mathbb{R}^n \). Matrix \( A \in \mathbb{R}^{m \times n} \) satisfies the restricted eigenvalue condition \( REC(S, \alpha, \beta) \) if the following holds for all \( x_1, x_2 \in S \):

\[
\alpha \|x_1 - x_2\|^2 \leq \|A(x_1 - x_2)\|^2 \leq \beta \|x_1 - x_2\|^2
\]

If \( \beta / \alpha \approx 1 \) then matrix \( A \) is a near-isometry on \( S \).

\( \delta \)-approximate projector: A network \( G(G^\dagger(\cdot)) : \mathbb{R}^n \to R(G) \) is called a \( \delta \)-approximate projector if the following holds for all \( x \in \mathbb{R}^n \):

\[
\|x - G(G^\dagger(x))\|^2 \leq \min_{z \in R^k} \|x - G(z)\|^2 + \delta
\]

- If \( \delta \approx 0 \) then this network projector is a near-ideal least-squares projector.
Convergence Guarantee

Assumptions:
• $A \in \mathbb{R}^{m \times n}$ satisfies $\text{REC}(S, \alpha, \beta), \beta/\alpha < 2$
• Concatenated network $G(G^\dagger(\cdot))$ is a $\delta$-approximate projector onto $\text{Range}(G)$
• $x^* \in R(G)$ and $y = Ax^*$
• NPGD is executed with step size $\eta = 1/\beta$

Guarantee:
• $f(x_n) \leq \left(\frac{\beta}{\alpha} - 1\right)^n f(x_0) + \frac{\beta \delta}{2-\beta/\alpha}$ Linear convergence
• After $n = \frac{1}{2-\beta/\alpha} \log \left(\frac{f(x_0)}{C \delta}\right)$ steps this algorithm achieves $\|x_n - x^*\|^2 \leq \left(C + \frac{1}{2\alpha/\beta - 1}\right) \delta$
• And as $n \to \infty$, $\|x^* - x_\infty\|^2 \leq \frac{\delta}{2\alpha/\beta - 1}$

- Error $\propto \delta$, convergence speed $\propto \alpha/\beta$
Learning Nice Sensing Matrices

- **NuMax:** find a measurement matrix that satisfies

  \[ REC(S, 1 - \delta_S, 1 + \delta_S) \]

  For training dataset of size \( M \), time complexity \( O(n^3 + M^4 n^2) \)

- **Kvinge et al:** heuristic iterative algorithm: find an orthonormal-row matrix with smallest \( \beta/\alpha \)

  Time complexity \( O(n^5) \) space complexity \( O(n^3) \)

- Faster algorithm?

---


Learning S-REC Matrices

Normalized secant set: for a given training set \( \mathcal{X} \), the normalized secant set is defined as:

\[
S(\mathcal{X}) = \left\{ \frac{x_1 - x_2}{\|x_1 - x_2\|} : x_1, x_2 \in \mathcal{X} \right\}
\]

For a given \( S(\mathcal{X}) \), optimize matrix \( A \in \mathbb{R}^{m \times n} \) with orthonormal rows for smaller \( \beta/\alpha \)

\[
\min_{A \in \mathbb{R}^{m \times n}} \frac{\beta}{\alpha} = \min_{A \in \mathbb{R}^{m \times n}} \frac{\max_{s \in S(\mathcal{X})} \|As\|^2}{\min_{s \in S(\mathcal{X})} \|As\|^2} \\
\leq \min_{AA^T = I_m} \frac{1}{\min_{s \in S(\mathcal{X})} \|As\|^2} \\
= \left( \max_{AA^T = I_m} \min_{s \in S(\mathcal{X})} \|As\|^2 \right)^{-1}
\]

- Requires iterating through all elements in \( S(\mathcal{X}) \).
- Only works on small secant set...
Learning S-REC Matrices

Normalized secant set: for a given training set $\mathcal{X}$, the normalized secant set is defined as:

$$S(\mathcal{X}) = \left\{ \frac{x_1 - x_2}{\|x_1 - x_2\|} : x_1, x_2 \in \mathcal{X} \right\}$$

For a given $S(\mathcal{X})$, optimize matrix $A \in \mathbb{R}^{m \times n}$ with orthonormal rows for smaller $\beta/\alpha$ : replace $\min_{s \in S(\mathcal{X})} \|As\|^2$ by $\frac{1}{M} \sum_{j=1}^{M} \|As_j\|^2$

Enables an efficient way to calculate $A^*$: construct $D = [s_1 \ | \ s_2 \ | \ldots \ | \ s_M]$

Compute EVD of $DD^T = \sum_{j=1}^{M} s_j s_j^T$

with time complexity $O(Mn^2 + n^3)$ and space complexity $O(n^2)$
S-REC for Various Inverse Problems

The S-REC value span for different inverse problems.

The tighter this span, the more stable the problem is.

A learned matrix clearly shrinks the s-REC span.
Learned and Random Matrix

MNIST NPGD reconstruction with learned matrix and with random Gaussian matrix.

A learned matrix is able to retain more information with few measurement (\(m\))

When \(m\) increases, the random matrix REC approaches that of a learned matrix
Results - Compressed Sensing

MNIST NPGD (our) reconstruction and comparison with other algorithms in compressed sensing setting. (A is Gaussian random matrix).
Compression ratio: 7.84
Results - Compressed Sensing

CelebA NPGD (our) reconstruction in compressed sensing.
(A is Gaussian random matrix).
Compression ratio: 12.29
Plots - Compressed Sensing (CelebA)

Left: Relative reconstruction error $\frac{\|\hat{x} - x\|^2}{\|x\|^2}$ with increasing $m$
Right: Structural similarity index (SSIM) with increasing $m$
Comparison: Run time

<table>
<thead>
<tr>
<th>$m$</th>
<th>CSGM $^2$</th>
<th>PGD-GAN</th>
<th>NPGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>5.8</td>
<td>66</td>
<td>0.09 (64x)</td>
</tr>
<tr>
<td>500</td>
<td>6.6</td>
<td>60</td>
<td>0.10 (66x)</td>
</tr>
<tr>
<td>1000</td>
<td>8.0</td>
<td>63</td>
<td>0.11 (72x)</td>
</tr>
<tr>
<td>2000</td>
<td>11.2</td>
<td>61</td>
<td>0.14 (80x)</td>
</tr>
</tbody>
</table>

Comparison of execution time ([sec.]) of recovery algorithms on the CelebA dataset. The relative speedup of our NPGD over the CSGM algorithm of Bora et al. is shown in parenthesis.
Image (LSUN-church) NPGD reconstruction under various inverse problems settings:
(A) Inpainting with 16x16 block (B) 2x super-resolution (C) Compressed sensing with compression ratio = 12.29
Conclusions

- Learning of regularizer independent of the imaging operator enables application to different operators without re-training
- Learning a generative model rather than a proximal map offers advantages
- Empirical estimation of REC
- Learned sensing matrix improves REC at high compression ratios
- Guaranteed performance - assuming successful training and REC
- Orders of magnitude speedup by eliminating inner optimization loop in previous GAN-based approaches.