Inverting Deep Generative Model, One Layer at a Time
Qi Lei¹, Ajil Jalal², Inderjit S. Dhillon¹,², and Alexandros G. Dimakis¹
¹University of Texas at Austin.   ²Amazon

Introduction

• We consider the inverse problem with a generator
  \[ G : \mathbb{R}^d \rightarrow \mathbb{R}^k, \]
  \[ z \leftarrow \arg \min_{x \in \mathbb{R}^d} \| x - AG(z) \|_2 \]
  (1)

• Applications
  • denoising
  • inpainting
  • reconstruction from Gaussian projections
  • phase retrieval
  • denoising

• Proximal Gradient Descent makes sure (1) is as hard as
  \[ \arg \min_{z \in \mathbb{R}^d} \| x - G(z) \|_2 \]
  (2)
  Therefore we focus on solving (2).

Setup

• A d-layer ReLU generative model:
  \[ G(z) = \text{ReLU}(W_d \cdots \text{ReLU}(W_2(\text{ReLU}(W_1z)))) \cdots ) \]

• Key concept: “ReLU Configuration”

Invertibility for Realizable Expansive ReLU Network: ReLU regression

• Expansive ReLU network:

Invertibility for Realizable ReLU Network: Hardness

• Inverting a single layer
  \[ w^\top_j z + b_j = x_j, \forall i \text{ s.t. } x_i > 0 \]
  \[ w^\top_j z + b_j \leq 0, \forall i \text{ s.t. } x_i = 0 \]

• Challenge for multiple layers: NP-complete problem

Theorem (NP-hardness to Recover ReLU Networks with Real Domain)

Given a four-layered ReLU neural network \( G(z) : \mathbb{R}^d \rightarrow \mathbb{R}^k \) where weights are all fixed, and an observation vector \( z \in \mathbb{R}^d \), the problem to determine whether there exists \( z \in \mathbb{R}^k \) such that \( G(z) = x \) is NP-complete.

• Challenge for multiple layers: non-convex pre-image (≥ 2 layers)

Experiments on Random Networks

• Network architecture: \( k \times 250 \times 600 \)
• Recovery with Various Input Dimension:

Experiments on Real Network for MNIST Dataset

• Network architecture: \( 20 \times 60 \times 784 \)
• Tasks: 1) Denoising, 2) Inpainting
• Noise generation: variance = 3e-1 Gaussian noise

Theorem (Exact Recovery for Random, Expansive and Realizable models)

Given a ReLU generative model (3) with random matrix and expansive factor \( c_0 \geq 2,1 \), and an observation \( x \in \mathbb{R}^k \), we are able to exactly recover \( z^* \in \mathbb{R}^k \) by conducting layer-wise linear regression (4), w.p. 1 - \( e^{-O(k)} \).

Invertibility for Noisy ReLU Networks

\( \ell_\infty \) Norm Error Bound: \( x = G(z) + e; \| e \|_\infty \leq \epsilon \)

• For a single layer, ground truth falls in:
  \[ x_j - \epsilon \leq w^\top_j z \leq x_j + \epsilon \quad \text{if } x_j > \epsilon, \quad j \in [n] \]
  \[ w^\top_j z \leq x_j + \epsilon \quad \text{if } x_j < \epsilon, \quad j \in [n] \]
  (5)

\( \ell_1 \) Norm Error Bound: \( x = G(z) + e; \| e \|_1 \leq \epsilon \)

• For a single layer, ground truth falls in:
  \[ x_j - \epsilon_j \leq w^\top_j z \leq x_j + \epsilon_j \quad \text{if } x_j > \epsilon \]
  \[ w^\top_j z \leq x_j + \epsilon_j \quad \text{if } x_j < \epsilon \]
  \[ \epsilon_j \geq 0, \sum \epsilon_j \leq \epsilon \]
  (6)

Theorem (\( \ell_\infty \) error bound)

Let \( x = G(z^*) + e \) be a noisy observation produced by the generator \( G \), and its weight matrix \( W_i \in \mathbb{R}^{(m_i, \infty)} \) is sampled from i.i.d Gaussian distribution \( \sim \mathcal{N}(0,1) \). Then there exists some constant \( c_0 \) as long as the error \( e \), \( \| e \|_\infty = \epsilon \), where \( \epsilon < \frac{c_0}{\sqrt{n}} \| z^* \|_2 \sqrt{\log \frac{\| z^* \|_2}{\epsilon}} \), such that by solving (5) recursively, we generate an \( z \) that satisfies \( \| z - z^* \|_\infty \leq \frac{c_0}{\sqrt{n}} \) w.h.p.

Figure: Success rate comparisons on random ReLU networks with different input dimension \( k \).

Figure: Recovery comparison using our algorithm \( \ell_\infty \) LP versus GD for an MNIST generative model.

Figure: Recovery comparison using non-identity sensing matrix using our algorithm \( \ell_\infty \) LP versus GD, for an MNIST generative model.

Time comparison

<table>
<thead>
<tr>
<th>k</th>
<th>10</th>
<th>30</th>
<th>50</th>
<th>70</th>
<th>90</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>( t_{\ell_\infty} ) LP</td>
<td>0.63</td>
<td>0.73</td>
<td>0.83</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>( \ell_1 ) LP</td>
<td>1.05</td>
<td>1.05</td>
<td>1.21</td>
<td>1.28</td>
<td>1.39</td>
<td>1.22</td>
</tr>
<tr>
<td>GD</td>
<td>1.59</td>
<td>1.65</td>
<td>1.72</td>
<td>1.80</td>
<td>2.09</td>
<td>2.01</td>
</tr>
</tbody>
</table>

Table: Comparison of CPU time cost averaged from 200 runs, including LP relaxation.

References
