Inverting Deep Generative Model, One Layer at a Time

Qi Lei[1], Ajil Jalal[1], Inderjit S. Dhillon[1,2], and Alexandros G. Dimakis[1]

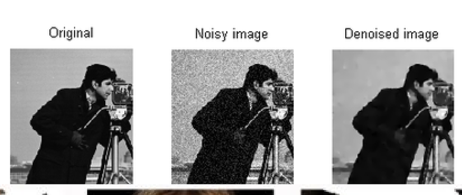
[1]University of Texas at Austin. [2]Amazon

Introduction

- We consider the inverse problem with a generator $G : \mathbb{R}^k \to \mathbb{R}^n$:

$$\boldsymbol{z} \leftarrow \arg\min_{\boldsymbol{z}\in\mathbb{R}^k} \|\boldsymbol{x} - AG(\boldsymbol{z})\|^2 \qquad (1)$$

- Applications
  - denoising



  - inpainting

  - reconstruction from Gaussian projections
  - phase retrieval
  - compression
- Proximal Gradient Descent makes sure (1) is as hard as

$$\arg\min_{\boldsymbol{z}\in\mathbb{R}^k} \|\boldsymbol{x} - G(\boldsymbol{z})\|^2 \qquad (2)$$

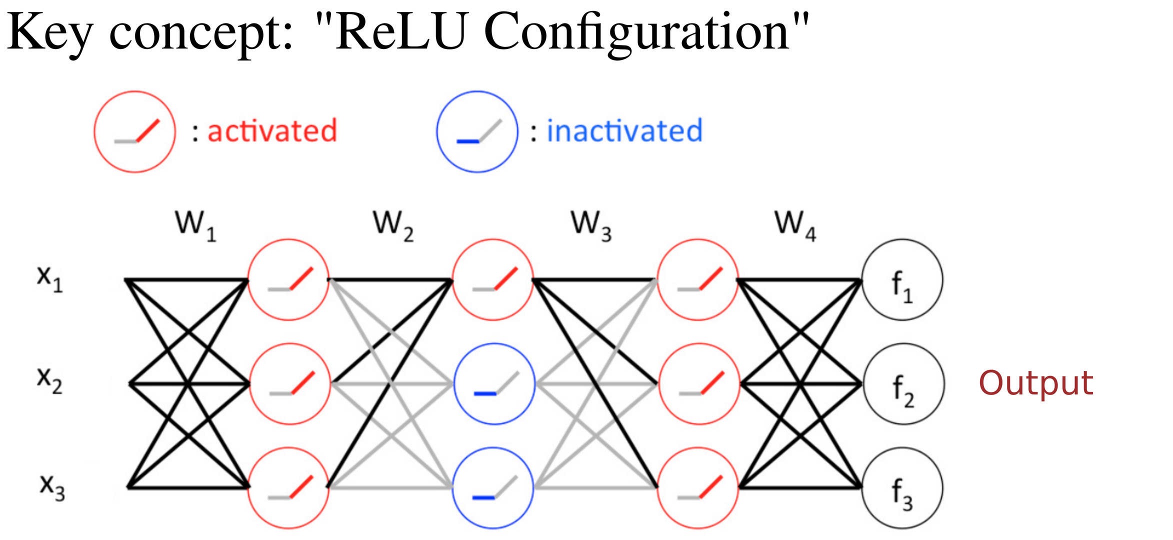
Therefore we focus on solving (2).

Setup

- A d-layer ReLU generative model:

$$G(\boldsymbol{z}) = \text{ReLU}(W_d \cdots \text{ReLU}(W_2(\text{ReLU}(W_1\boldsymbol{z}))\cdots)), \qquad (3)$$

- Key concept: "ReLU Configuration"



Invertibility for Realizable ReLU Network: Hardness

- Inverting a single layer

$$\boldsymbol{w}_i^\top \boldsymbol{z} + b_i = x_i, \forall i \text{ s.t. } x_i > 0$$
$$\boldsymbol{w}_i^\top \boldsymbol{z} + b_i \le 0, \forall i \text{ s.t. } x_i = 0 \qquad (4)$$

- Challenge for multiple layers: NP-complete problem

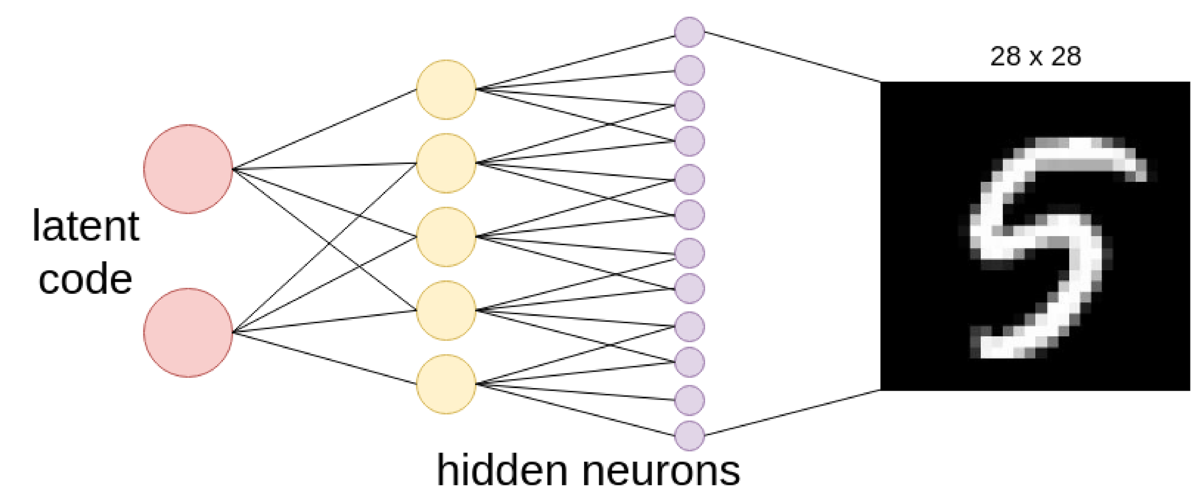
Theorem (NP-hardness to Recover ReLU Networks with Real Domain)

Given a four-layered ReLU neural network $G(\boldsymbol{x}) : \mathbb{R}^k \to \mathbb{R}^2$ where weights are all fixed, and an observation vector $x \in \mathbb{R}^2$, the problem to determine whether there exists $\boldsymbol{z} \in \mathbb{R}^k$ such that $G(\boldsymbol{z}) = \boldsymbol{x}$ is NP-complete.

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

Invertibility for Realizable Expansive ReLU Network: ReLU regression

- Expansive ReLU network:



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

Given a ReLU generative model (3) with random matrix and expansive factor $c_0 \ge 2.1$, and an observation $\boldsymbol{x} \in \mathbb{R}^n$, we are able to exactly recover $\boldsymbol{z}^* \in \mathbb{R}^k$ by conducting layer-wise linear regression (4), w.p $1 - e^{-\Omega(k)}$.

Invertibility for Noisy ReLU Networks

ℓ_∞ Norm Error Bound: $\boldsymbol{x} = G(\boldsymbol{z}) + \boldsymbol{e}, \|\boldsymbol{e}\|_\infty \le \epsilon$

- For a single layer, ground truth falls in:

$$x_j - \epsilon \le \boldsymbol{w}_j^\top \boldsymbol{z} \le x_j + \epsilon \text{ if } x_j > \epsilon, j \in [n]$$
$$\boldsymbol{w}_j^\top \boldsymbol{z} \le x_j + \epsilon \text{ if } x_j \le \epsilon, j \in [n], \qquad (5)$$

Theorem (ℓ_∞ error bound)

Let $\boldsymbol{x} = G(\boldsymbol{z}^*) + \boldsymbol{e}$ be a noisy observation produced by the generator G, such that its weight matrix $W_i \in \mathbb{R}^{n_{i-1}\times n_i}$ ($n_i \ge 5n_{i-1}, \forall i$) is *sampled from i.i.d Gaussian distribution* $\sim \mathcal{N}(0,1)$. Then there exists some constant c_2, as long as the error $\boldsymbol{e}, \|\boldsymbol{e}\|_\infty = \epsilon$, where $\epsilon < \frac{c_2^d}{2^{d+1}}\|\boldsymbol{z}^*\|_2\sqrt{k}$, such that by solving (5) recursively, we generate an $\boldsymbol{z}$ that satisfies $\|\boldsymbol{z} - \boldsymbol{z}^*\|_\infty \le \frac{2^d\epsilon}{c_2^d}$ w.h.p.

ℓ_1 Norm Error Bound: $\boldsymbol{x} = G(\boldsymbol{z}) + \boldsymbol{e}, \|\boldsymbol{e}\|_1 \le \epsilon$

- For a single layer, ground truth falls in:

$$x_j - e_j \le \boldsymbol{w}_j^\top \boldsymbol{z} \le x_j + e_j, \text{ if } x_j > \epsilon$$
$$\boldsymbol{w}_j^\top \boldsymbol{z} \le x_j + e_j, \text{ if } x_j \le \epsilon$$
$$e_j \ge 0, \sum_j e_j \le \epsilon \qquad (6)$$

Theorem (ℓ_1 error bound)

Let $\boldsymbol{x} = G(\boldsymbol{z}^*) + \boldsymbol{e}$ be a noisy observation produced by the generator G, and its weight matrix $W_i \in \mathbb{R}^{n_{i-1}\times n_i}$ satisfy (m_i, ∞)-RIP-1 with the integer $m_i > n_{i-1}$ and constant c_1. Let $\epsilon := \|\boldsymbol{e}\|_1$, and suppose each observation $\boldsymbol{z}_i$ at each layer has at least m_i coordinates are larger than $\frac{2^{d+1-i}\epsilon}{c_1^{d-i}}$. Then by recursively solving (6), it produces a $\boldsymbol{z}$ that satisfies $\|\boldsymbol{z} - \boldsymbol{z}^*\|_1 \le \frac{2^d\epsilon}{c_1^d}$ w.h.p.

Experiments on Random Networks

- Network architecture: $k \times 250 \times 600$
- Recovery with Various Input Dimension:

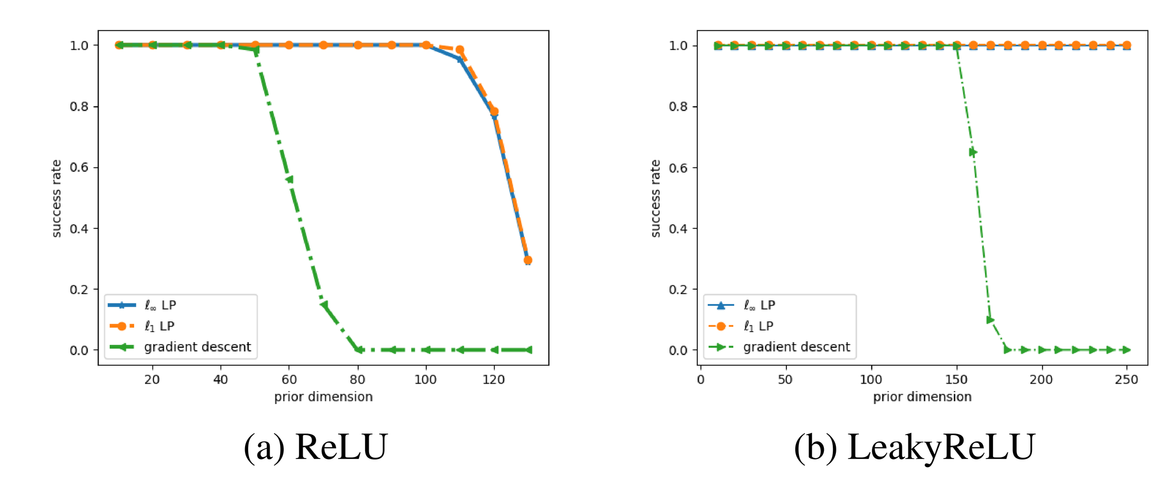


Figure: Success rate comparisons on random ReLU networks with different input dimension k.

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

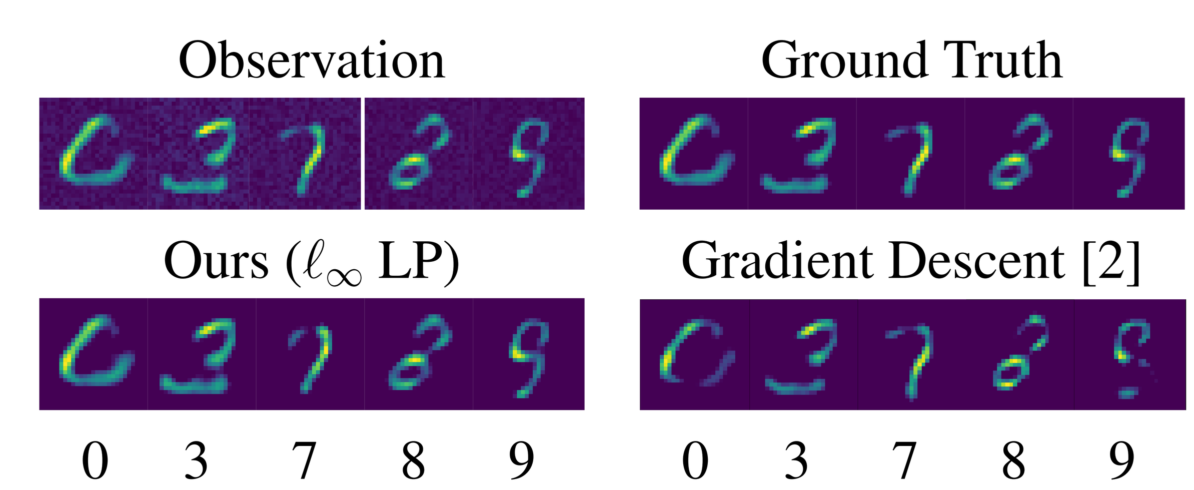


Figure: Recovery comparison using our algorithm ℓ_∞ LP versus GD for an MNIST generative model.

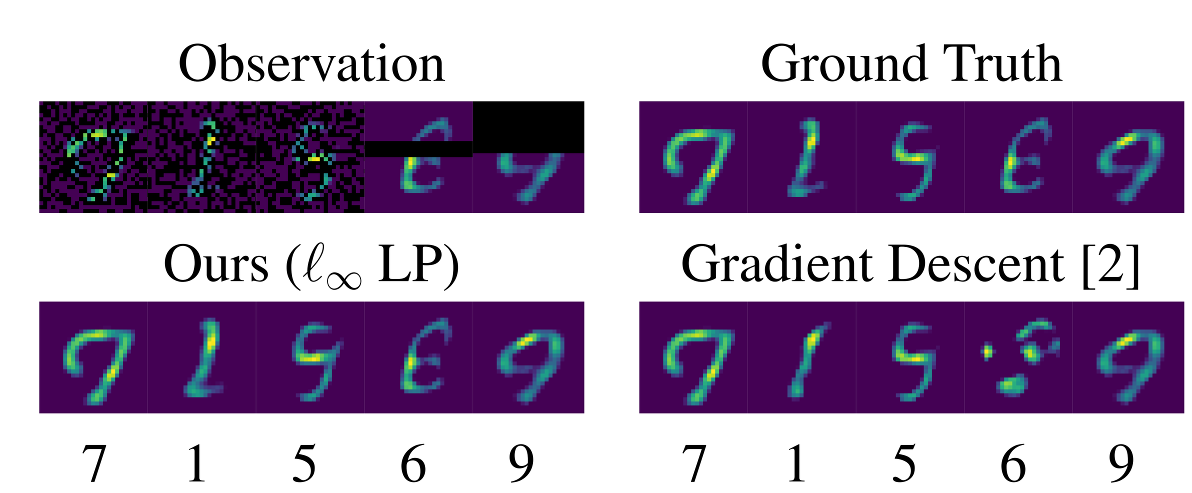


Figure: Recovery comparison with non-identity sensing matrix using our algorithm ℓ_∞ LP versus GD, for an MNIST generative model.

Time comparison

k	10	30	50	70	90	110	MNIST
ℓ_∞ LP	0.63	0.73	0.83	0.90	0.95	1.03	0.5
ℓ_1 LP	1.05	1.05	1.23	1.28	1.39	1.22	1.1
GD	1.59	1.65	1.72	1.80	2.09	2.01	72

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

References

[1] 1. Bora, Ashish, et al. "Compressed sensing using generative models." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

[2] 2.Hand, Paul, and Vladislav Voroninski. "Global guarantees for enforcing deep generative priors by empirical risk." arXiv preprint arXiv:1705.07576 (2017).