

Recovery Guarantees of Unsupervised Neural Networks for Inverse Problems trained with Gradient Descent

Nathan Buskalic, Yvain Quéau and Jalal Fadili
Normandie Univ., UNICAEN, ENSICAEN, CNRS, GREYC
Caen, France
nathan.buskalic@unicaen.fr

Abstract—Advanced machine learning methods, and more prominently neural networks, have become standard to solve inverse problems over the last years. However, the theoretical recovery guarantees of such methods are still scarce and difficult to achieve. Only recently did unsupervised methods such as the Deep Image Prior (DIP) get equipped with convergence and recovery guarantees for generic loss functions when trained through gradient flow with an appropriate initialization. In this paper, we extend these results by proving that these guarantees hold true when using gradient descent with an appropriately chosen step-size/learning rate. We also show that the discretization only affects the overparametrization bound for a two-layer DIP network by a constant and thus that the different guarantees found for the gradient flow will hold for gradient descent.

Index Terms—Inverse problems, Deep Image/Inverse Prior, Overparametrization, Gradient descent, Unsupervised learning.

I. INTRODUCTION

Problem statement. In finite dimension, inverse problems are understood as the task of reliably recovering a vector $\bar{\mathbf{x}}$ in a finite-dimensional vector space (throughout \mathbb{R}^n) from its indirect and noisy measurements

$$\mathbf{y} = \mathbf{A}\bar{\mathbf{x}} + \varepsilon, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^m$ is the observation vector, $\mathbf{A} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a forward operator that we assume linear and ε is an additive noise. We will use the shorthand notation $\bar{\mathbf{y}} \stackrel{\text{def}}{=} \mathbf{A}\bar{\mathbf{x}}$ to denote the noiseless observation vector.

Due to the large number of scientific and engineering fields where inverse problems appear, it is natural that in recent years sophisticated machine learning algorithms, including those based on (deep) neural networks, were developed to solve them. These methods have shown promising results; for space limitation, we refer to the reviews [1], [2]. Many of these approaches are based on the idea of optimizing a generator network $\mathbf{g} : (\mathbf{u}, \boldsymbol{\theta}) \in \mathbb{R}^d \times \mathbb{R}^p \mapsto \mathbf{x} \in \mathbb{R}^n$, equipped with an activation function ϕ , to transform an input (latent) variable $\mathbf{u} \in \mathbb{R}^d$ into a vector \mathbf{x} as close as possible to the sought-after vector. The optimization/training takes the form of a (possibly stochastic) gradient descent on the parameters $\boldsymbol{\theta}$ of the network to minimize a loss function $\mathcal{L}_{\mathbf{y}} : \mathbb{R}^m \rightarrow \mathbb{R}_+$, $\mathbf{v} \mapsto \mathcal{L}_{\mathbf{y}}(\mathbf{v})$ intended to capture the forward model (1) by measuring the discrepancy between the observation \mathbf{y} and an estimated

observation $\mathbf{v} = \mathbf{A}\mathbf{g}(\mathbf{u}, \boldsymbol{\theta})$ generated by the network with parameters/weights $\boldsymbol{\theta}$.

Literature overview. Several theoretical works emerged recently to study the optimization trajectory of overparametrized networks [3], [4]. While the first attempts used unrealistic assumptions, such as the strong convexity of the loss when composed with the network, the next attempts were based on gradient dominated inequalities. This allowed to prove, for networks trained to minimize the mean square error (MSE), an exponential convergence rate to an optimal solution of either gradient flow [5], [6], or gradient descent [7], [8], [9], [10], [11]). These results were generalized in [12] to the inverse problem setting, providing both convergence and recovery guarantees. Furthermore, the latter work alleviates the restriction of using the MSE as the loss function and gives results for generic loss functions obeying a globalized Kurdyka-Łojasiewicz (KL) inequality (e.g., any convex semi-algebraic or definable function) [13], [14], [15]. However, these guarantees were only provided for the gradient flow and it is yet unknown how forward Euler discretization of this flow, giving rise to gradient descent, affects them.

Contributions. Gradient descent iteration to learn the network parameters at iteration τ reads

$$\boldsymbol{\theta}_{\tau+1} = \boldsymbol{\theta}_{\tau} - \gamma \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \boldsymbol{\theta}_{\tau})), \quad (2)$$

with γ the (fixed) descent-size/learning rate. The goal of this work is to analyze under which conditions the iterates (2) converge, whether they converge to a zero-loss solution, and what can be said about the recovery guarantees of $\bar{\mathbf{x}}$.

In Section III, we first show that neural networks trained through gradient descent (2) can benefit from convergence and recovery guarantees for general loss functions verifying a globalized KL property. That is, we prove that under a proper initialization and a well-chosen step-size γ , the network will converge to a zero-loss solution at a rate dependent on the desingularizing function of the KL property of the loss. We also provide a bound on the recovery error of the original vector $\bar{\mathbf{x}}$ which requires a restricted injectivity condition to hold, and we emphasize the trade-off between this condition and the expressivity of the trained network. Then, we give a bound on the overparametrization necessary for a two-layer DIP [16] network to benefit from all these guarantees

with high probability. Our results match those of the gradient flow up to discretization errors that depend on the step-size. Section IV eventually provides numerical experiments validating our theoretical findings.

II. PRELIMINARIES

A. General Notations

For a matrix $\mathbf{M} \in \mathbb{R}^{a \times b}$ we denote by $\sigma_{\min}(\mathbf{M})$ and $\sigma_{\max}(\mathbf{M})$ its smallest and largest non-zero singular values, and by $\kappa(\mathbf{M}) = \frac{\sigma_{\max}(\mathbf{M})}{\sigma_{\min}(\mathbf{M})}$ its condition number. We use as a shorthand $\sigma_{\mathbf{A}}$ to express $\sigma_{\min}(\mathbf{A})$. We denote by $\|\cdot\|$ the Euclidian norm of a vector. With a slight abuse of notation $\|\cdot\|$ will also denote the spectral norm of a matrix. We use the notation $a \gtrsim b$ if there exists a constant $C > 0$ such that $a \geq Cb$. We also define $\mathbf{x}_\tau = \mathbf{g}(\mathbf{u}, \boldsymbol{\theta}_\tau)$ and recall that $\mathbf{y}_\tau = \mathbf{A}\mathbf{x}_\tau$. The Jacobian operator of $\mathbf{g}(\mathbf{u}, \cdot)$ is denoted $\mathcal{J}_{\mathbf{g}}$. $\mathcal{J}_{\mathbf{g}}(\tau)$ is a shorthand notation of $\mathcal{J}_{\mathbf{g}}$ evaluated at $\boldsymbol{\theta}_\tau$. The local Lipschitz constant of a mapping on a ball of radius $R > 0$ around a point \mathbf{z} is denoted $\text{Lip}_{\mathbb{B}(\mathbf{z}, R)}(\cdot)$. We omit R in the notation when the Lipschitz constant is global. For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we use the notation for the sublevel set $[f < c] = \{\mathbf{z} \in \mathbb{R}^n : f(\mathbf{z}) < c\}$ and $[c_1 < f < c_2] = \{\mathbf{z} \in \mathbb{R}^n : c_1 < f(\mathbf{z}) < c_2\}$. We set $C_\phi = \sqrt{\mathbb{E}_{X \sim \mathcal{N}(0,1)} [\phi(X)^2]}$ and $C_{\phi'} = \sqrt{\mathbb{E}_{X \sim \mathcal{N}(0,1)} [\phi'(X)^2]}$.

For some $\Theta \subset \mathbb{R}^p$, we define $\Sigma_\Theta = \{\mathbf{g}(\mathbf{u}, \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$ the set of vectors that the network $\mathbf{g}(\mathbf{u}, \cdot)$ can generate for all $\boldsymbol{\theta}$ in the set of parameters Θ . Σ_Θ can thus be viewed as a parametric manifold. If Θ is closed (resp. compact), so is Σ_Θ . We denote $\text{dist}(\cdot, \Sigma_\Theta)$ the distance to Σ_Θ which is well defined if Θ is closed and non-empty. For a vector \mathbf{x} , $\mathbf{x}_{\Sigma_\Theta}$ is its projection on Σ_Θ , i.e. $\mathbf{x}_{\Sigma_\Theta} \in \text{Argmin}_{\mathbf{z} \in \Sigma_\Theta} \|\mathbf{x} - \mathbf{z}\|$. We also define $T_{\Sigma_\Theta}(\mathbf{x})$ the tangent cone of Σ_Θ at $\mathbf{x} \in \Sigma_\Theta$. The minimal (conic) singular value of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ w.r.t. the cone $T_{\Sigma_\Theta}(\mathbf{x})$ is then defined as

$$\lambda_{\min}(\mathbf{A}; T_{\Sigma_\Theta}(\mathbf{x})) = \inf\{\|\mathbf{A}\mathbf{z}\| / \|\mathbf{z}\| : \mathbf{z} \in T_{\Sigma_\Theta}(\mathbf{x})\}.$$

Throughout, $\mathbf{g}(\mathbf{u}, \boldsymbol{\theta})$ is a feedforward neural network.

Definition II.1. Let $\phi : \mathbb{R} \rightarrow \mathbb{R}$ be a component wise activation function. An L -layer fully connected neural network is a collection of weight matrices $(\mathbf{W}^{(l)})_{l \in [L]}$ where $\mathbf{W}^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$ with $N_l \in \mathbb{N}$ the number of neurons on layer l . We take $\boldsymbol{\theta} \in \mathbb{R}^p$, with $p = \sum_{l=1}^L N_{l-1}N_l$, a vector that gathers all these parameters. Then, a neural network parametrized by $\boldsymbol{\theta}$ is the mapping $\mathbf{g} : (\mathbf{u}, \boldsymbol{\theta}) \in \mathbb{R}^d \times \mathbb{R}^p \mapsto \mathbf{g}(\mathbf{u}, \boldsymbol{\theta}) \in \mathbb{R}^{N_L}$ with $N_L = n$, which is defined recursively as

$$\begin{cases} \mathbf{g}^{(0)}(\mathbf{u}, \boldsymbol{\theta}) &= \mathbf{u}, \\ \mathbf{g}^{(l)}(\mathbf{u}, \boldsymbol{\theta}) &= \phi(\mathbf{W}^{(l)}\mathbf{g}^{(l-1)}(\mathbf{u}, \boldsymbol{\theta})), \quad l = 1, \dots, L-1, \\ \mathbf{g}(\mathbf{u}, \boldsymbol{\theta}) &= \mathbf{W}^{(L)}\mathbf{g}^{(L-1)}(\mathbf{u}, \boldsymbol{\theta}). \end{cases}$$

B. KL inequality

We will work under a general condition of the loss function $\mathcal{L}_{\mathbf{y}}$. We will suppose that $\mathcal{L}_{\mathbf{y}}$ verifies a globalized version of the Kurdyka-Łojasewicz-type (KL for short) inequality [15, Theorem 1].

Definition II.2 (KL inequality). A continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with $\min f = 0$ satisfies the KL inequality if there exists $r_0 > 0$ and a strictly increasing function $\psi \in \mathcal{C}^0([0, r_0[) \cap \mathcal{C}^1(]0, r_0])$ with $\psi(0) = 0$ such that

$$\psi'(f(\mathbf{z})) \|\nabla f(\mathbf{z})\| \geq 1, \quad \text{for all } \mathbf{z} \in [f < r_0]. \quad (3)$$

We use the shorthand notation $f \in \text{KL}_\psi(r_0)$ for a function satisfying this inequality.

III. RECOVERY GUARANTEES WITH GRADIENT DESCENT

A. Main Assumptions

Throughout this paper, we will work under the following standing assumptions:

A-1. $\mathcal{L}_{\mathbf{y}}(\cdot) \in \mathcal{C}^1(\mathbb{R}^m)$ is bounded from below whose gradient is Lipschitz continuous on the bounded sets of \mathbb{R}^m .

A-2. $\mathcal{L}_{\mathbf{y}}(\cdot) \in \text{KL}_\psi(\mathcal{L}_{\mathbf{y}}(\mathbf{y}_0) + \eta)$ for some $\eta > 0$.

A-3. $\phi \in \mathcal{C}^1(\mathbb{R})$ and $\exists B > 0$ such that $\sup_{x \in \mathbb{R}} |\phi'(x)| \leq B$ and ϕ' is B -Lipschitz continuous.

A-4. $(\boldsymbol{\theta}_\tau)_{\tau \in \mathbb{N}}$ is bounded.

Notably, these assumptions are not restrictive. A-1 and A-3 ensure that $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$ is locally Lipschitz w.r.t $\boldsymbol{\theta}$ and A-2 is met by many classical loss functions (MSE, Kullback-Leibler and cross entropy to cite a few). A-4 is quite mild as we do not require neither convexity nor coercivity of the objective or that $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$ is globally Lipschitz continuous. Indeed, the latter is widely assumed to ensure that the scheme (2) has the descent property, but this is unrealistic when training neural networks. Our assumption A-4, together with A-1 and A-3, ensure the existence of a constant $L > 0$ such that $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$ is L -Lipschitz continuous w.r.t $\boldsymbol{\theta}$ on the ball containing $(\boldsymbol{\theta}_\tau)_{\tau \in \mathbb{N}}$. This avoids an ‘‘egg and chicken’’ issue. Indeed, we can show that $(\boldsymbol{\theta}_\tau)_{\tau \in \mathbb{N}}$ is bounded if (2) has a global descent property, but this requires global Lipschitz continuity of $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$. A-4 resolves that issue.

B. Deterministic Results

We can now state our recovery theorem for gradient descent. For obvious space limitation, the proof can be found in [17].

Theorem III.1. Consider a network $\mathbf{g}(\mathbf{u}, \cdot)$, a forward operator \mathbf{A} and a loss $\mathcal{L}_{\mathbf{y}}$ such that our assumptions hold. Let $(\boldsymbol{\theta}_\tau)_{\tau \in \mathbb{N}}$ be the sequence generated by (2). There exists a constant $L > 0$ such that if $\gamma \in]0, 1/L]$ and if the initialization $\boldsymbol{\theta}_0$ is such that

$$\sigma_{\min}(\mathcal{J}_{\mathbf{g}}(0)) > 0 \text{ and } R' < R \quad (4)$$

where R' and R obey

$$R' = \frac{2\nu_1\psi(\mathcal{L}_{\mathbf{y}}(\mathbf{y}_0))}{\sigma_{\mathbf{A}}\sigma_{\min}(\mathcal{J}_{\mathbf{g}}(0))} \text{ and } R = \frac{\sigma_{\min}(\mathcal{J}_{\mathbf{g}}(0))}{2\text{Lip}_{\mathbb{B}(\boldsymbol{\theta}_0, R)}(\mathcal{J}_{\mathbf{g}})}, \quad (5)$$

with $\nu_1 = \frac{1+\gamma L}{1-\gamma L/2} \in]1, 4]$, then the following holds:

(i) the loss converges to 0 at the rate

$$\mathcal{L}_{\mathbf{y}}(\mathbf{y}_\tau) \leq \Psi^{-1}(\xi_\tau) \quad (6)$$

with Ψ a primitive of $-(\psi')^2$ and $\xi_\tau = \frac{\sigma_{\mathbf{A}}^2 \sigma_{\min}(\mathcal{J}_{\mathbf{g}}(0))^2}{4\nu_2} \tau + \Psi(\mathcal{L}_{\mathbf{y}}(\mathbf{y}_0))$ where $\nu_2 = \frac{(1+\gamma L)^2}{(\gamma-\gamma^2 L/2)}$. Moreover, $(\boldsymbol{\theta}_\tau)_{\tau \in \mathbb{N}}$ converges to a global minimizer $\boldsymbol{\theta}_\infty$ of $\mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$, at the rate

$$\|\boldsymbol{\theta}_\tau - \boldsymbol{\theta}_\infty\| \leq \frac{2\nu_1 \psi(\Psi^{-1}(\xi_\tau))}{\sigma_{\min}(\mathcal{J}_{\mathbf{g}}(0))\sigma_{\mathbf{A}}}; \quad (7)$$

(ii) if $\text{Argmin}(\mathcal{L}_{\mathbf{y}}(\cdot)) = \{\mathbf{y}\}$, $\mathcal{L}_{\mathbf{y}}$ is convex and

A-5. $\ker(\mathbf{A}) \cap T_{\Sigma'}(\bar{\mathbf{x}}_{\Sigma'}) = \{0\}$ with $\Sigma' \stackrel{\text{def}}{=} \Sigma_{\mathbb{B}_{R'+\|\boldsymbol{\theta}_0\|}}$
Then,

$$\begin{aligned} \|\mathbf{x}_\tau - \bar{\mathbf{x}}\| &\leq \frac{\psi(\Psi^{-1}(\xi_\tau))}{\lambda_{\min}(\mathbf{A}; T_{\Sigma'}(\bar{\mathbf{x}}_{\Sigma'}))} \\ &+ \left(1 + \frac{\|\mathbf{A}\|}{\lambda_{\min}(\mathbf{A}; T_{\Sigma'}(\bar{\mathbf{x}}_{\Sigma'}))}\right) \text{dist}(\bar{\mathbf{x}}, \Sigma') \\ &+ \frac{\|\boldsymbol{\varepsilon}\|}{\lambda_{\min}(\mathbf{A}; T_{\Sigma'}(\bar{\mathbf{x}}_{\Sigma'}))}. \end{aligned} \quad (8)$$

C. Discussion and Consequences

We start by discussing the conditions of the theorem. The first condition $R' < R$ ensures that the loss at initialization is “small enough” so that the parameters $\boldsymbol{\theta}_0$ lie in the attraction basin of a minimizer. The condition $\gamma \in]0, 1/L[$ is classical and ensures that the scheme (2) has a descent property. In our context, avoiding big steps guarantees that it will not step out of the attraction basin of the minimizer.

Let us now comment on the different claims of the theorem. The first one ensures that the network converges to a zero-loss solution with a rate dictated by the mapping Ψ^{-1} (which is decreasing) applied to an affine increasing function of τ , validating the need for an early-stopping strategy to avoid overfitting the noise. The choice of the loss function is generally dictated by a fidelity argument to the forward model (1) (e.g., noise $-\log$ -likelihood). On the other hand, Ψ only depends on the KL desingularizing function of the chosen loss function. Thus, this choice not only influences the theoretical convergence rate but also the condition (4) required for the theorem to hold. Our second claim shows that gradient descent provides a sequence of network parameters that converges to a global minimum of the loss. On the signal recovery side, our third result gives us a reconstruction error bound that holds under a restricted injectivity constraint. This bound depends on both the expressivity of the network (via the set Σ') and decreases with decreasing noise. Observe that a more expressive network, i.e., larger Σ' , yields a lower $\text{dist}(\bar{\mathbf{x}}, \Sigma')$ but may hinder the restricted injectivity condition A-5. The recovery bound (8) also suggests to use an early stopping strategy to ensure that $\mathbf{x}(\tau)$ for τ large enough (whose expression is left to the reader), will lie in a ball around $\bar{\mathbf{x}}$.

Theorem III.1 shows that gradient descent, which is nothing but an Euler forward time discretization of the gradient flow, inherits the recovery properties of latter proved in [12]. The price to be paid by gradient descent is in the requirement that the sequence of iterates a priori bounded, that γ is well-chosen, as well as by the constants ν_1 and ν_2 . As $\nu_1 > 1$ and strictly

increasing in γ , condition $R' < R$ in (4) is strictly worse than its counterpart for gradient flow, and the error bound (7) is also strictly larger. On the other hand, the convergence behaviour gets faster with decreasing ν_2 . If γ gets close to 0, ν_1 will get smaller but ν_2 will grow very large, entailing a very slow convergence. In fact, ν_2 is minimized when $\gamma = \frac{1}{2L}$ in which case $\nu_2 = 6L$. As such, this choice of the step-size incurs a trade-off between the convergence speed and the requirements of the theorem.

D. Probabilistic Bounds For a Two-Layer DIP Network

We now study the case of a two-layer network in the DIP setting. We fix the latent input \mathbf{u} and learn the weights of a network to match the observation \mathbf{y} . We consider the two-layer network defined as

$$\mathbf{g}(\mathbf{u}, \boldsymbol{\theta}) = \frac{1}{\sqrt{k}} \mathbf{V} \phi(\mathbf{W}\mathbf{u}) \quad (9)$$

with $\mathbf{V} \in \mathbb{R}^{n \times k}$ and $\mathbf{W} \in \mathbb{R}^{k \times d}$ and ϕ an activation function obeying A-3. We will assume the following:

A-6. \mathbf{u} is a uniform vector on \mathbb{S}^{d-1} ;

A-7. \mathbf{W}_0 has iid entries from $\mathcal{N}(0, 1)$ and $C_\phi, C_{\phi'} < +\infty$ with $C_\phi = \sqrt{\mathbb{E}_{X \sim \mathcal{N}(0,1)} [\phi(X)^2]}$ and $C_{\phi'} = \sqrt{\mathbb{E}_{X \sim \mathcal{N}(0,1)} [\phi'(X)^2]}$;

A-8. \mathbf{V}_0 is independent from \mathbf{W}_0 and \mathbf{u} and has iid columns with identity covariance and D -bounded centered entries.

Our main result gives a bound on the level of overparametrization which is sufficient for (4) to hold with high probability.

Corollary III.2. *Suppose that assumptions A-1 and A-3 hold. Let C, C' two positive constants that depend only on the activation function and D. Let:*

$$L_{\mathcal{L},0} = \max_{\mathbf{v} \in \mathbb{B}(0, C\|\mathbf{A}\| \sqrt{n \log(d)} + \sqrt{m}(\|\mathbf{A}\bar{\mathbf{x}}\|_\infty + \|\boldsymbol{\varepsilon}\|_\infty))} \frac{\|\nabla_{\mathbf{v}} \mathcal{L}_{\mathbf{y}}(\mathbf{v})\|}{\|\mathbf{v} - \mathbf{y}\|}.$$

Consider the one-hidden layer network (9) where both layers are trained with gradient descent using initialization satisfying A-6 to A-8 and the architecture parameters obeying

$$\begin{aligned} k \geq C' \sigma_{\mathbf{A}}^{-4} n \psi \left(\frac{L_{\mathcal{L},0}}{2} \left(C \|\mathbf{A}\| \sqrt{n \log(d)} \right. \right. \\ \left. \left. + \sqrt{m} (\|\mathbf{A}\bar{\mathbf{x}}\|_\infty + \|\boldsymbol{\varepsilon}\|_\infty) \right)^2 \right)^4. \end{aligned}$$

Then (4) holds with probability at least $1 - 2n^{-1} - d^{-1}$.

The overparametrization bound is strongly dependent on the chosen loss function via the associated desingularizing function ψ . As an example, for the MSE loss, we have $\psi(s) = cs^{1/2}$. In that setting, the dependency on the problem variables becomes $k \gtrsim \kappa(\mathbf{A})n^2m$ which indicates naturally that the conditioning of the operator plays an important role in the reconstruction capabilities of the network. Furthermore, the dimension of the signal is more impactful on the bound than the dimension of the observation which seems intuitive as the network tries to reconstruct a vector in \mathbb{R}^n while matching

a vector in \mathbb{R}^m . This bound matches the one from the gradient flow case as the discretization error is absorbed in the constant. A more in-depth discussion can be found in [12].

Local Lipschitz constant estimate. Estimating the local Lipschitz constant L of $\nabla_{\theta} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$ is not easy in general. One can still derive crude bounds which can give information on the behaviour of L . To give some guidelines, and make the discussion easier, we here focus on the MSE loss and assume that ϕ is uniformly bounded (this conforms to our numerics). In this case, we actually have global L -Lipschitz continuity $\nabla_{\theta} \mathcal{L}_{\mathbf{y}}(\mathbf{A}\mathbf{g}(\mathbf{u}, \cdot))$ and one can show that

$$L \lesssim \|\mathbf{A}\|^2 \frac{n}{k} + \|\mathbf{A}\| \|\mathbf{y} - \mathbf{y}_0\| \sqrt{\frac{n}{k}}, \quad (10)$$

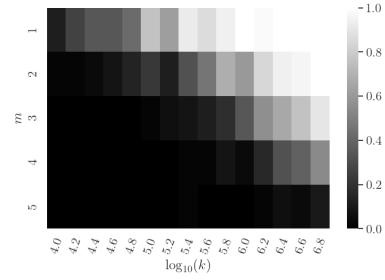
where the constant in \lesssim depends only on B, D and the bound on ϕ . As the level of overparametrization increases, the bound on L gets smaller and one can afford taking larger step-size γ . On the other hand, for the overparametrized regime, L depends essentially on $\|\mathbf{A}\|$ and the loss at initialization, which may themselves depend on the dimensions (m, n) , indicating that the choice of γ may be influenced by (m, n) , regardless of the choice of k large enough.

IV. NUMERICAL EXPERIMENTS

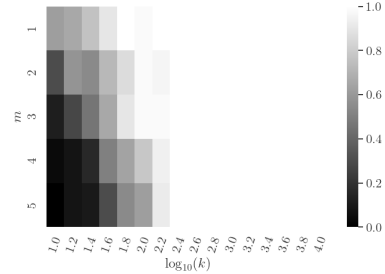
We trained two-layer networks, equipped with the sigmoid activation, with varying architectures and on problems with varying dimensions. In the first experiments we set $n = 5$, and sample iid entries of \mathbf{A} and $\bar{\mathbf{x}}$ from the standard Gaussian distribution. Our first experiment aims at verifying numerically the level of overparametrization needed for condition (4) to hold and compare it with the one that yields convergence to a zero loss. In this experiment, we use the MSE loss and only train the hidden layer of the network as discussed in [12]. From Figure 1 we see that our theoretical bound in Corollary III.2 is validated, though it is pessimistic as our analysis is a worst-case one. For instance, for $m = 2$, our bound is quite conservative as it requires $k \geq 10^{6.8}$ while in practice, convergence occurs when $k \geq 10^{2.5}$.

Our second experiment depicted in Figure 2 aims at verifying experimentally the proper choice of γ for gradient descent to converge to an optimal solution with different values n and fixed $k = 10^4$. For the tested range of n , the network is overparameterized and thus $n/k = O(1)$ in (10). One can see from Figure 2 that there is a clear correlation between the dimension of the problem n and the allowed choice of γ . Indeed, by concentration of Gaussian matrices, we have for any $\delta > 0$, $\|\mathbf{A}\| \leq \sqrt{n} + (1 + \delta)\sqrt{m}$ with high probability. Thus, according to (10), γ must get smaller as n increases. This is confirmed by another experiment that we do not show here for obvious space limitation reasons, where we have observed that when we fix n and m and vary k , there is a threshold effect where, whatever k is chosen, the network training will start to diverge when $\gamma \geq 10^{1.6}$, validating the earlier discussion.

For our next experiment, we will illustrate how our models behave on images taken from the tiny ImageNet dataset, that we consider as vectors in $[0, 255]^{4096}$. We used $k = 10^4$ as it was empirically sufficient to achieve convergence. In



(a) Empirical probability over 50 runs that (4) is verified as a function of m and k .



(b) Empirical probability over 50 runs to converge to an optimal solution as a function of m and k , i.e., where $\mathcal{L}_{\mathbf{y}}(\mathbf{y}_{10^5}) \leq 10^{-14}$.

Fig. 1: Level of overparametrization needed for (4) to hold compared to the one required to converge in practice.

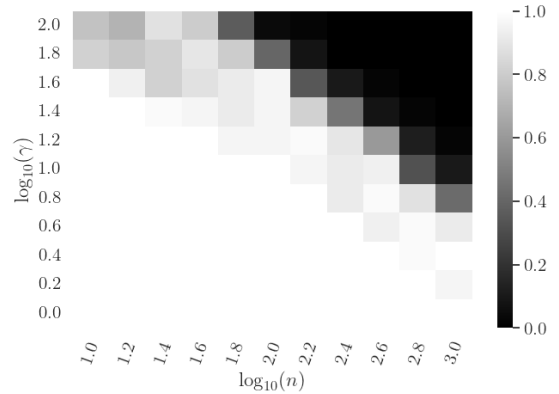
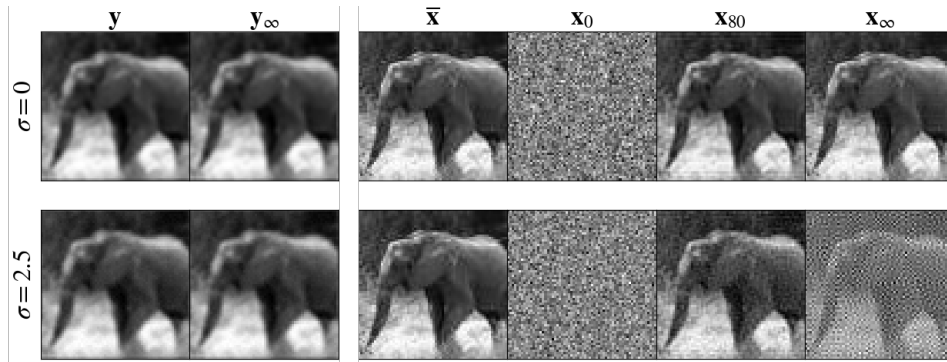
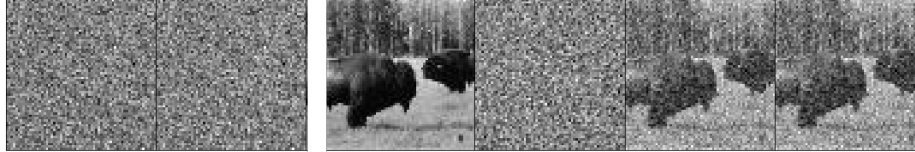


Fig. 2: Probability over 50 runs for a network to converge to an optimal solution for various n and γ .

Figure 3a, \mathbf{A} is a convolution with a Gaussian kernel of standard deviation 1 (thus, $n = m$ here), and tested two scenarios: noiseless and one with low level of an additive zero-mean white Gaussian noise (AWGN) with standard deviation 2.5. In the noiseless case, the original image is perfectly recovered. The presence of noise, even small, entails a degraded reconstruction in the image space, though the reconstruction is very good in the observation space. This is predicted by our theoretical results due to the fact that the blur operator, while being injective, is very badly conditioned, that is $\sigma_{\min}(\mathbf{A}) \sim 10^{-5}$, hence greatly amplifying the noise in the reconstruction (see (8)). An early stopping strategy is thus necessary to avoid overfitting the noise.



(a) Reconstruction of an image when one uses Gaussian blur with no noise and with low level of noise.



(b) Evolution of the reconstruction through training with a well-conditioned operator and high level of noise.

Fig. 3: Deep Inverse Prior applied to image reconstruction.

In Figure 3b, we changed \mathbf{A} to a better conditioned one. For this, we generated two random orthogonal matrices, a diagonal matrix with entries evenly spaced between 1 and 2, and then formed \mathbf{A} whose SVD is these three matrices. We used an AWGN with standard deviation 50, entailing that the observation is overwhelmed by noise (see left image of Figure 3b). Figure 3b also displays the reconstructed images obtained at increasing iterations of gradient descent training. We see that the reconstructed image gets better after a few iterations, before converging to a solution. To confirm these visual results, we report in Figure 4 the evolution of the reconstruction error $\|\mathbf{x}_\tau - \bar{\mathbf{x}}\|$ vs the iteration τ , where the upper-bound predicted by our theorem is also shown in dashed line. We observe that indeed the recovery error decreases until $\tau \approx 10$ and then increases slightly before stabilizing.

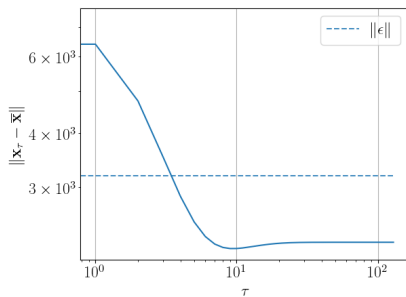


Fig. 4: Evolution of the reconstruction error with a well-conditioned operator and high amount of noise.

REFERENCES

- [1] S. Arridge, P. Maass, Ö. Ozan, and C-B. Schönlieb, “Solving inverse problems using data-driven models,” *Acta Numerica*, vol. 28, pp. 1–174, 2019.
- [2] G. Ongie, A. Jalal, C. A Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, “Deep learning techniques for inverse problems in imaging,” *IEEE JSAT*, vol. 1, no. 1, pp. 39–56, 2020.
- [3] P. L. Bartlett, A. Montanari, and A. Rakhlin, “Deep learning: a statistical viewpoint,” *Acta numerica*, vol. 30, pp. 87–201, 2021.
- [4] C. Fang, H. Dong, and T. Zhang, “Mathematical models of overparameterized neural networks,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 683–703, 2021.
- [5] L. Chizat, E. Oyallon, and F. Bach, “On lazy training in differentiable programming,” *NeurIPS*, vol. 32, 2019.
- [6] D. AR. Robin, K. Scaman, et al., “Convergence beyond the overparameterized regime using rayleigh quotients,” *NeurIPS*, vol. 35, pp. 10725–10736, 2022.
- [7] S. Du, X. Zhai, B. Póczos, and A. Singh, “Gradient Descent Provably Optimizes Over-parameterized Neural Networks,” in *ICLR*, 2019.
- [8] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, “Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks,” in *ICML*, 2019, pp. 322–332.
- [9] S. Oymak and M. Soltanolkotabi, “Overparameterized nonlinear learning: Gradient descent takes the shortest path?,” in *ICML*, 2019, pp. 4951–4960.
- [10] S. Oymak and M. Soltanolkotabi, “Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks,” *IEEE JSAT*, vol. 1, no. 1, pp. 84–105, 2020.
- [11] C. Liu, L. Zhu, and M. Belkin, “Loss landscapes and optimization in over-parameterized non-linear systems and neural networks,” *Applied and Computational Harmonic Analysis*, vol. 59, pp. 85–116, 2022.
- [12] N. Buskulic, J. Fadili, and Y. Quéau, “Convergence and recovery guarantees of unsupervised neural networks for inverse problems,” *JMIV*, pp. 1–22, 2024.
- [13] S. Łojasiewicz, “Une propriété topologique des sous-ensembles analytiques réels,” *Coll. du CNRS, Les équations aux dérivées partielles*, vol. 117, no. 87-89, pp. 2, 1963.
- [14] S. Łojasiewicz, “Sur les trajectoires du gradient d’une fonction analytique,” *Semin. Geom., Univ. Studi Bologna*, vol. 1982/1983, pp. 115–117, 1984.
- [15] K. Kurdyka, “On gradients of functions definable in o-minimal structures,” in *Annales de l’institut Fourier*, 1998, vol. 48, pp. 769–783, Issue: 3.
- [16] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.
- [17] N. Buskulic, J. Fadili, and Y. Quéau, “Recovery guarantees of unsupervised neural networks for inverse problems trained with gradient descent,” *Hal preprint 04496435*, 2024.