Abstract—Rectified linear units, or ReLUs, have become a preferred activation function for artificial neural networks. In this paper we consider two basic learning problems assuming that the underlying data follow a generative model based on a simple network with ReLU activations. The first problem we study corresponds to learning a generative model in the presence of nonlinearity (modeled by the ReLU functions). Given a set of signal vectors $y^i \in \mathbb{R}^d$, $i=1,2,\ldots,n$, we aim to learn the network parameters, i.e., the $d \times k$ matrix $A$, under the model $y^i = \text{ReLU}(Ac^i + b)$, where $b \in \mathbb{R}^d$ is a random bias vector. We show that it is possible to recover the column space of $A$ within an error of $O(d)$ (in Frobenius norm) under certain conditions on the distribution of $b$.

The second problem we consider is that of robust recovery of the signal in the presence of outliers. In this setting, we are interested in recovering the latent vector $c$ from its noisy nonlinear images of the form $v = \text{ReLU}(Ac) + e + w$, where $e \in \mathbb{R}^d$ denotes the outliers with sparsity $s$ and $w \in \mathbb{R}^d$ denote the dense but small noise. We show that the LASSO algorithm recovers $c \in \mathbb{R}^k$ within an $\ell_2$-error of $O(\sqrt{(k + s) \log d}/d)$ when $A$ is a random Gaussian matrix.

Index Terms—ReLU networks, robust signal recovery, matrix estimation, generative models, dictionary learning.

I. INTRODUCTION

Rectified Linear Unit (ReLU) is a basic nonlinear function defined to be $\text{ReLU} : \mathbb{R} \to \mathbb{R}_+ \cup \{0\}$ as $\text{ReLU}(x) = \max(0,x)$. For any matrix $X$, $\text{ReLU}(X)$ denotes the matrix obtained by applying the ReLU function on each of the coordinates of the matrix $X$. ReLUs are building blocks of many nonlinear data-fitting problems based on deep neural networks (see, e.g., [1] for a good exposition). In particular, [2] showed that supervised training of very deep neural networks is much faster if the hidden layers are composed of ReLUs.

Let $\mathcal{Y} \subset \mathbb{R}^d$ be a collection of signal vectors that are of interest to us. Depending on the application at hand, the signal vectors, i.e., the constituents of $\mathcal{Y}$, may range from images, speech signals, network access patterns to user-item rating vectors and so on. We assume that the signal vectors satisfy a generative model [3]–[5], where each signal vector can be approximated by a map $g : \mathbb{R}^k \to \mathbb{R}^d$ from the latent space to the ambient space, i.e., for each $y \in \mathcal{Y}$,

$$y \approx g(c) \text{ for some } c \in \mathbb{R}^k.$$  \hfill (1)

In this paper we consider the following specific model (single layer ReLU-network), with the weight (generator) matrix $A \in \mathbb{R}^{d \times k}$ and bias $b \in \mathbb{R}^d$:

$$y = \text{ReLU}(Ac + b).$$  \hfill (2)

The generative model in (2) raises multiple interesting questions that play fundamental role in understanding the underlying data and designing systems and algorithms for information processing. Two such most basic questions are as follows:

1. **Learning the network parameters:** Given the $n$ observations $\{y^i\}_{i \in [n]} \subset \mathbb{R}^d$ from the model (cf. (2)), recover the parameters of the model, i.e., $A \in \mathbb{R}^{d \times k}$ such that

$$y^i = \text{ReLU}(Ac^i + b) \quad \forall \ i \in [n],$$  \hfill (3)

with latent vectors $\{c^i\}_{i \in [n]} \subset \mathbb{R}^k$. We assume that the bias vector $b$ is a random vector comprising of i.i.d. coordinates with each coordinate distributed according to the probability density function $p(\cdot)$. This question is closely related to the dictionary-learning problem [6]. We also note that this question is different from training a model (such as, [7]), in which case the set $\{c^i\}_{i \in [n]}$ is known (and possibly chosen accordingly). We define the $d \times n$ observation matrix $Y = [y^1 \ y^2 \ \cdots \ y^n]$. Similarly, we define the $k \times n$ coefficient matrix $C = [c^1 \ c^2 \ \cdots \ c^n]$. With this notion, we can concisely represent the $n$ observation vectors as

$$Y = \text{ReLU} \left( AC + b \otimes 1^T \right) = \text{ReLU} \left( M + b \otimes 1^T \right),$$  \hfill (4)

where $M = AC$ and $1 \in \mathbb{R}^n$ denotes the all-ones vector.

2. **Recovery of the signal in the presence of errors:** Given the erroneous (noisy) version of a vector generated by the model (cf. (2)), denoise the observation or recover the latent vector. Formally, given

$$v = y + e + w = \text{ReLU}(Ac + b) + e + w,$$  \hfill (5)

and the knowledge of model parameters, obtain $\hat{c} \in \mathbb{R}^k$ such that $\|e - \hat{e}\|$ is small\footnote{A related problem of denoising in the presence of outliers only focuses on obtaining an estimate $\hat{y}$ which is close to the true signal vector $y$.}. In the aforementioned noisy observation,
\( w \in \mathbb{R}^d \) denotes the (dense) noise vector with bounded norm. On the other hands, the vector \( e \in \mathbb{R}^d \) contains (potentially) large corruptions, also referred to as sparse errors or outliers (we assume, \( \|e\|_0 \leq s \)).

For this part, we focus on the setting where the weight matrix \( A \) is a random Gaussian matrix with i.i.d. entries. Furthermore, another crucial assumption is that the outlier vector is oblivious in its nature\(^2\), i.e., \( e \) is not picked in an adversarial manner given the knowledge of \( A \) and \( c \).

Apart from being closely related, one of our main motivations (we assume, \( c \)) is the problem of single layer \( A \times C \) from \( Y \) with an error in Frobenius norm at most \( O(\sqrt{d}) \) with high probability (see Theorem 3 for the details).

\(^2\)It is an interesting problem to extend our results to a setting with adversarial errors. However, we note that this problem is an active area of research even in the case of linear measurement, i.e. \( y = Ac + e + w \) [8], [9]. We plan to explore this problem in future work.

Apart from being closely related, one of our main motivations (we assume, \( c \)) is the problem of single layer \( A \times C \) from \( Y \) with an error in Frobenius norm at most \( O(\sqrt{d}) \) with high probability (see Theorem 3 for the details).

However both of these works do not consider the presence of outliers (sparse but large noise) in the observation. The sparse noise is quite natural to assume; for instance, many times only partial observations of a signal vector are obtained. Analyzing any learning algorithm in the presence of outliers is both practically and theoretically important, as demonstrated by a large body of literature on the treatment of outliers in other domains. Naturally, a real-life signal may significantly deviate from what the underlying model predicts, and the ability to handle outliers enables inference with such signals. Similarly, the data collection may be error-prone (e.g., due to unreliable source, faulty storage/transmission) which may lead to the observed signal having a significant number of outliers. Such errors cannot be modeled with small noise. Furthermore, one of our motivations – associative memories – require performing signal recovery in the presence of outliers. The ReLU model with outliers as considered in this paper can be thought of as a nonlinear version of the problem of recovering \( e \) from linear observations of the form \( y = Ac + e \), with \( e \) denoting the outliers. This problem with linear observations was studied in the celebrated work of [16]. We note that the technique of [16] does not extend to the case when there is a dense (but bounded) noise component present. Our result in this case is a natural generalization and complementary to the one in [1] in that 1) we present a recovery method which is robust to outliers, and 2) instead of analyzing the gradient descent algorithm, we directly analyze the performance of the minimizer of our optimization program.

a) Related works in network parameter learning: There have been a recent surge of interest in learning ReLUs, and the above two questions are of basic interest even for a single-layer network (i.e., nonlinearity comprising of a single ReLU function). It is conceivable that understanding the behavior of a single-layer network would allow one to use some iterative peeling off technique to develop a theory for multiple layers.

To the best of our knowledge, the problem of learning network parameters, even for single-layer networks has not been studied as such, i.e., theoretical guarantees do not exist. The principled approaches to solve this unsupervised problem in practice reduce this to the ‘training’ problem, such as the autoencoders [13] that learn features by extensive end-to-end training of encoder-decoder pairs; or use the recently popular generative adversarial networks (GAN) [3] that utilize a discriminator network to tune the generative network. The method that we are going to propose here can be seen as an alternative to using GANs for this purpose, and can be seen as an isolated ‘decoder’ learning of the autoencoder. Note that the problem bears some similarity with matrix completion problems, a fact we greatly exploit. In matrix completion, a matrix \( M \) is visible only partially, and the task is to recover the unknown entries by exploiting some prior knowledge about \( M \). In the case of (3), we are more likely to observe the positive entries of \( M \), which, unlike a majority of matrix completion literature, creates the dependence between \( M \) and the sampling procedure.

b) Related works in signal recovery: In [14], the recovery problem of single layer ReLU-model under the reliable agnostic learning model of [15] was considered. Informally speaking, under very general distributional assumptions (the rows of \( A \) are sampled from some distribution), given \( A \) and \( y = \text{ReLU}(Ac) \), [14] proposes an algorithm that recovers a hypothesis which has an error-rate (under some natural loss function defined therein) of \( \epsilon \) with respect to the true underlying ReLU-model. Moreover, the algorithm runs in time polynomial in \( d \) and exponential in \( 1/\epsilon \). As opposed to this, given \( A \) and the corresponding output of the ReLU-network \( Y = \text{ReLU}(Ac + b) \), we focus on the problem of recovering \( c \) itself. In [1], further results on this model under somewhat different learning guarantees were obtained. Assuming that the entries of the matrix \( A \) to be i.i.d. Gaussian, [1] shows that with high probability a gradient descent algorithm recovers \( c \) within some precision in terms of \( \ell_2 \)-loss: the relative error decays exponentially with the number of steps of the algorithm. The obtained result is more general as it extends to constrained optimizations in the presence of some regularizers (e.g., \( e \) can be restricted to be a sparse vector).

c) Main result for learning network parameters: We assume to have observed \( d \times n \) matrix \( Y = \text{ReLU}(AC + b \otimes 1^T) \), where \( A \) and \( C \) are unknown \( d \times k \) and \( k \times n \) matrices, respectively. Moreover, \( b \in \mathbb{R}^d \) is a random i.i.d. bias, and \( \otimes \) denotes the Kronecker product. Note that this model ensures that the bias corresponding to each coordinate is random, but does not change over different signal vectors. We show that the maximum-likelihood method guarantees the recovery of the matrix \( AC \) from \( Y \) with an error in Frobenius norm at most \( O(\sqrt{d}) \) with high probability (see Theorem 3 for the
formal statement). Then leveraging the well known results on matrix perturbation [17], it is possible to also recover the column space of $A$ with the similar guarantee.

The main technique that we use to obtain this result is inspired by the work on matrix completion by Davenport et al. [18]. One of the main challenges that we face here is that while an entry of the matrix $Y$ is a random variable (since $b$ is a random bias), whether that is being observed or being cut-off by the ReLU function (for being negative) depends on the value of the entry itself. In general matrix completion literature, the entries of the matrix being observed are sampled independent of the underlying matrix itself (see, e.g., [19]–[21] and references therein). For this reason, we cannot use most of these results off-the-shelf. However, similar predicament is (partially) present in [18], where entries are quantized while being observed.

That said, our observation model differs from [18] in a critical way: in our case the bias vector, while random, does not change over observations. This translates to less freedom during the transformation of the original matrix to the observed matrix, leading to dependence among the elements in a row. Furthermore, the analysis becomes notably different since the positive observations are not quantized.

d) Main result for noisy recovery: We plan to recover $c \in \mathbb{R}^k$ from observations $v = \text{ReLU}(Ae + b) + e + w$, where $A$ is a $d \times k$ standard i.i.d. Gaussian matrix, $e \in \mathbb{R}^d$ is the vector containing outliers (sparse noise) with sparsity $\|e\|_0 \leq s$, and $w \in \mathbb{R}^d$ is bounded dense noise such that $\|w\|_{\infty} \leq \delta$. To recover $c$ and $e$ we employ the LASSO algorithm, which is inspired by the work of [22] and [23]. In particular, Plan and Vershynin recently showed that a signal can be provably recovered (up to a constant multiple) from its nonlinear Gaussian measurements via the LASSO algorithm by treating the measurements as linear observations [22]. In the context of ReLU model, for outlier-free measurements $v = \text{ReLU}(Ae + b) + w$, it follows from [22] that LASSO algorithm outputs $\mu e$ as the solution with $\mu = \mathbb{E}(g \cdot \text{ReLU}(g + b))$, where $g$ is a Gaussian random variable and $b$ is a random variable denoting bias associated with the ReLU function. We show that this approach guarantees with high probability recovery of $c$ within an $\ell_2$-error of $O(\sqrt{(k + s) \log d})$ even when the measurements are corrupted by the outliers $e$. This is achieved by jointly minimizing the square loss over $(c, e)$ after treating our measurements as linear measurements $v' = Ac + e + w$ and adding an $\ell_1$-regularizer to the loss function to promote the sparsity of the solution for $e$ (we also recover $c$, see Theorem 4 for a formal description).

e) Organization: In Section II, we describe some of the notations used throughout the paper and introduce the technical tools that would be useful to prove our main results. In Section III, we present our main result on the network parameter learning problem. Section IV contains our contribution for the recovery problem in the presence of outliers.

II. NOTATIONS AND TECHNICAL TOOLS

For any positive integer $n$, define $[n] \equiv \{1, 2, \ldots, n\}$. Given a matrix $M \in \mathbb{R}^{d \times n}$, for $(i, j) \in [d] \times [n]$, $M_{i,j}$ denotes the $(i,j)$-th entry of $M$. For $i \in [d]$, $m_i = (M_{i1}, M_{i2}, \ldots, M_{in})^T$ denotes the vector containing the elements of the $i$-th row of $M$. Similarly, for $j \in [n]$, $m^j = (M_{1j}, M_{2j}, \ldots, M_{dj})^T$ denotes the $j$-th column of $M$. Recall that the function $\text{ReLU} : \mathbb{R} \to \mathbb{R}_+ \cup \{0\}$ takes the following form $\text{ReLU}(x) = \max(x, 0)$. For a matrix $M \in \mathbb{R}^{d \times n}$, we use $\text{ReLU}(M)$ to denote the $d \times n$ matrix obtained by applying the ReLU function on each entry of $M$. For two matrix $A$ and $B$, we use $A \otimes B$ to represent the Kronecker product of $A$ and $B$. Given a matrix $M \in \mathbb{R}^{d \times n}$, $\|M\|_F = \sqrt{\sum_{i,j} M_{ij}^2}$ denotes its Frobenius norm. Also, let $\|M\|$ denote the $\ell_2$-operator norm of $M$, i.e. the maximum singular value of $M$. We use $\|M\|_*$ to denote the nuclear norm of $M$.

We define the following flatness parameter associated with a function $f : \mathbb{R} \to \mathbb{R}$:

$$\beta_{\alpha}(f) := \inf_{|x| \leq \alpha} \frac{|f'(x)|^2}{4f(x)}.$$  \hfill (6)

This parameter quantifies how flat $f$ can be in the interval $[-\alpha, \alpha]$. We also define a Lipschitz parameter $L_\alpha(f)$ for $f$ as follows:

$$L_\alpha(f) := \max \left\{ \sup_{|x| \leq \alpha} \frac{f(x)}{\int_{-\infty}^x f(y) dy}, \sup_{|x| \leq \alpha} \frac{f'(x)}{f(x)} \right\}.$$  \hfill (7)

In the course of this paper, namely in the network parameter learning part, we use the key tools of symmetrization and contraction to bound the supremum of an empirical process following the lead of [18] and the analysis of generalization bounds in the statistical learning literature [24]. In particular, we need the following two results.

Theorem 1 (Symmetrization of expectation [25]). Let $X_1, X_2, \ldots, X_d$ be $n$ independent RVs taking values in $\mathcal{X}$ and $F$ be a class of $\mathcal{F}$-valued functions on $\mathcal{X}$. Furthermore, let $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d$ be $d$ independent Rademacher RVs. Then, for any $t \geq 1$,

$$\mathbb{E} \left( \sup_{f \in \mathcal{F}} \sum_{i=1}^d \left( f(X_i) - \mathbb{E} f(X_i) \right) \right)^t \leq 2^t \cdot \mathbb{E} \left( \sup_{f \in \mathcal{F}} \sum_{i=1}^d \varepsilon_i f(X_i) \right)^t.$$  \hfill (8)

Theorem 2 (Contraction principle [25]). Let $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_d$ be $d$ independent Rademacher RVs and $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a convex and increasing function. Let $\zeta : \mathbb{R} \to \mathbb{R}$ be an $L$-Lipschitz function, i.e., $|\zeta(a) - \zeta(b)| \leq L|a - b|$, which satisfy $\zeta(0) = 0$. Then, for any $T \subseteq \mathbb{R}^d$,

$$\mathbb{E} f \left( \frac{1}{2} \sup_{t \in T} \sum_{i=1}^d \varepsilon_i \zeta(t_i) \right) \leq \mathbb{E} f \left( L \cdot \sup_{t \in T} \sum_{i=1}^d \varepsilon_i t_i \right).$$
III. LEARNING PARAMETERS IN A SINGLE-LAYER RELU-NETWORK

In the paper, we employ the natural approach to learn the underlying weight matrix $A$ from the observation matrix $Y$. As the network maps a lower dimensional latent vector $c \in \mathbb{R}^k$ to obtain a signal vector $y = \text{ReLU}(Ac + b)$ in dimension $d > k$, the matrix $M = AC$ (cf. (4)) is a low-rank matrix as long as $k < \min\{d, n\}$. In our quest of recovering the weight matrix $A$, we first focus on estimating the matrix $M$, when given access to $Y$. This task can be viewed as estimating a low-rank matrix from its partial (randomized) observations. Our approach is inspired by the recent work of [18] on 1-bit matrix completion. However, the crucial difference between our model and the model of [18] arises due to the fact that the 1) bias vector $b$ does not change over observations in our case, which lead to more complicated situation with dependent random variables, 2) our observations are not quantized.

Furthermore, although our formulation is close to [18], because of the aforementioned differences in the observation models, we get much stronger guarantee on the recovery of the matrix. Indeed, our results are comparable to analogous results of [26], [27] that also study the quantized matrix completion problem.

A. Learning from rectified observations

We now focus on the task of estimating $M$ from the observation matrix $Y$ (cf. (4)). For $i \in [d]$, we define $N_Y(i) \subseteq [n]$ as the set of positive coordinates of the $i$-th row of the matrix $Y$, i.e.,

$$N_Y(i) = \{j \in [n] : Y_{i,j} > 0\} \quad \text{and} \quad N_{Y,i} = |N_Y(i)|.$$

Note that, for $i \in [d]$, the original matrix $M$ needs to satisfy

$$M_{i,j} + b_i = Y_{i,j} \quad \text{for} \quad j \in N_Y(i) \quad \text{(9)}$$

and

$$M_{i,j} + b_i < 0 \quad \text{for} \quad j \in \overline{N_Y(i)} := [n]\setminus N_Y(i). \quad \text{(10)}$$

For $i \in [d]$ and $j \in [n]$, let $M_{i,(j)}$ denote the $j$-th largest element of the $i$-th row of $M$, i.e., for $i \in [d]$, $M_{i,(1)} \geq M_{i,(2)} \geq \cdots \geq M_{i,(n)}$. It is straightforward to verify from (9) that $N_Y(i)$ denotes the indices of $N_{Y,i}$ largest entries of the $i$-th row of $M$. Furthermore, with $N_{Y,i} \in [n]$, we have

$$b_i = Y_{i,(1)} - M_{i,(1)} = \cdots = Y_{i,(N_{Y,i})} - M_{i,(N_{Y,i})}. \quad \text{(11)}$$

Similarly, it follows from (10) that whenever we have $N_{Y,i} = 0$, then $b_i$ belongs to the interval $(-\infty, -\max_{j\in[n]} M_{i,(j)}) = (-\infty, -M_{i,(1)})$. Based on these observation, we define the set of matrices $\mathcal{X}_{Y,\nu,\gamma} \subseteq \mathbb{R}^{d \times n}$ as

$$\mathcal{X}_{Y,\nu,\gamma} = \{ X : \|X\|_{\infty} \leq \gamma; \ Y_{i,(1)} - X_{i,(2)} = \cdots = Y_{i,(N_{Y,i})} - X_{i,(N_{Y,i})}; \text{ and} \ X_{i,(N_{Y,i})} \geq \max_{j \in \overline{N_Y(i)}} X_{i,j} + \nu \forall i \in [d] \}. \quad \text{(12)}$$

Recall that, $p : \mathbb{R} \to \mathbb{R}$ denote the probability density function of each bias RV. Thus, the likelihood that a matrix $X \in \mathcal{X}_{Y,\nu,\gamma}$ results into the observation matrix $Y$ is as follows.

$$\mathbb{P}(Y|X) = \prod_{i \in [d]} \mathbb{P}(y_i|x_i), \quad \text{(13)}$$

where, for $i \in [d], \mathbb{P}(y_i|x_i) = \mathbb{I}_{\{N_{Y,i} = 0\}} \cdot \mathbb{P}(b_i \leq -\max_{j\in[n]} X_{i,j})$$

$$+ \sum_{s=1}^n \mathbb{I}_{\{N_{Y,i} = s\}} \cdot p(b_i = Y_{i,(s)} - X_{i,(s)}). \quad \text{(14)}$$

By using the notation $F(x_1, x_2) = \mathbb{P}(-x_1 \leq B \leq -x_2)$ and $X^*_i = \max_{j\in[n]} X_{i,j}$, we can rewrite (14) as follows.

$$\mathbb{P}(y_i|x_i) = \mathbb{I}_{\{N_{Y,i} = 0\}} \cdot F(\infty, X^*_i)$$

$$+ \sum_{s=1}^n \mathbb{I}_{\{N_{Y,i} = s\}} \cdot \log p(Y_{i,(s)} - X_{i,(s)}). \quad \text{(15)}$$

Therefore, the log-likelihood of observing $Y$ given that $X$ is the original matrix is the following.

$$\mathcal{L}_Y(X) = \sum_{i \in [d]} \log \mathbb{P}(y_i|x_i)$$

$$= \sum_{i \in [d]} \left( \mathbb{I}_{\{N_{Y,i} = 0\}} \cdot \log F(\infty, X^*_i) \right. \left. + \sum_{s=1}^n \mathbb{I}_{\{N_{Y,i} = s\}} \cdot \log \frac{p(Y_{i,(s)} - X_{i,(s)})}{p(Y_{i,(s)})} \right). \quad \text{(16)}$$

In what follows, we work with a slightly modified quantity

$$\mathcal{Z}_Y(X) = \mathcal{L}_Y(X) - \mathcal{L}_Y(0).$$

Note that

$$\mathcal{Z}_Y(X) = \sum_{i \in [d]} \left( \mathbb{I}_{\{N_{Y,i} = 0\}} \cdot \log \frac{F(\infty, X^*_i)}{F(\infty, 0)} \right. \left. + \sum_{s=1}^n \mathbb{I}_{\{N_{Y,i} = s\}} \cdot \log \frac{p(Y_{i,(s)} - X_{i,(s)})}{p(Y_{i,(s)})} \right). \quad \text{(17)}$$

In order to recover the matrix $M$ from the observation matrix $Y$, we employ the natural maximum likelihood approach which is equivalent to the following.

$$\text{maximize}_{X \in \mathbb{R}^{d \times n}} \quad \mathcal{Z}_Y(X) \quad \text{subject to} \quad X \in \mathcal{X}_{Y,\nu,\gamma}. \quad \text{(18)}$$

Define $\omega_{p,\gamma,\nu}$ to be such that $F(x, y) \geq \omega_{p,\gamma,\nu}$ for all $x, y \in [-\gamma, \nu]$ with $|x - y| > \nu$. In what follows, we simply refer this quantity as $\omega_p$ as $\gamma$ and $\nu$ are clear from context. The following result characterizes the performance of the program proposed in (18).

**Theorem 3.** Assume that $\|M\|_{\infty} \leq \gamma$ and the observation matrix $Y$ is related to $M$ according to (4). Let $\tilde{M}$ be the solution of the program specified in (18), and the bias density function $p(x)$ is differentiable with bounded derivative. Then, the following holds with probability at least $1 - \frac{1}{4+n}$:

$$\|M - \tilde{M}\|_F^2 \leq C_0 L_\gamma(p) \cdot \frac{\gamma d}{\beta_\gamma(p) \omega_p}, \quad \text{(19)}$$
where, $C_0$ is a constant. The quantities $\beta_\gamma(p)$ and $L_\gamma(p)$ depend on the distribution of the bias and are defined in (6) and (7), respectively.

The proof of Theorem 3 crucially depends on the following lemma. We refer the reader to [28, Appendix B] for a detailed proof of the lemma.

**Lemma 1 (Defining the distance measure).** Given the observation matrix $Y$ which is related to the matrix $M$ according to (4), let $X_{Y,\nu,\gamma}$ be as defined in (12). Then, for any $X \in X_{Y,\nu,\gamma}$, we have

\[
E \left[ |\mathcal{Z}_Y(M) - \mathcal{Z}_Y(X)| \right] \geq \beta_\gamma(p) \omega_p \cdot \|M - X\|_F^2.
\]  
(20)

Now we are ready to prove Theorem 3 by utilizing Lemma 1.

**B. Proof of Theorem 3**

Let $\hat{M}$ be the solution of the program in (18). In what follows, we use $\mathcal{X}$ as a short hand notation for $X_{Y,\nu,\gamma}$. We have,

\[
0 \leq \mathcal{Z}_Y(\hat{M}) - \mathcal{Z}_Y(M) = E \left[ \mathcal{Z}_Y(\hat{M}) - \mathcal{Z}_Y(M) \right] -
\]

\[
(\mathcal{Z}_Y(M) - E [\mathcal{Z}_Y(M)]) + (\mathcal{Z}_Y(\hat{M}) - E [\mathcal{Z}_Y(M)])
\]

\[
\leq E \left[ \mathcal{Z}_Y(\hat{M}) - \mathcal{Z}_Y(M) \right] + 2 \sup_{X \in \mathcal{X}} \mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)],
\]

which implies that

\[
E \left[ \mathcal{Z}_Y(M) - \mathcal{Z}_Y(\hat{M}) \right] \leq 2 \sup_{X \in \mathcal{X}} \mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)].
\]

We now employ Lemma 1 to obtain that

\[
\beta_\gamma(p) \omega_p \|M - \hat{M}\|_F^2 \leq 2 \sup_{X \in \mathcal{X}} \mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)].
\]

(21)

We now proceed to upper bound the right hand side of (21). It follows from the standard symmetrization trick (cf. Theorem 1) that, for any integer $t \geq 1$, we have

\[
\mathbb{E} \left[ \sup_{X \in \mathcal{X}} |\mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)]|^t \right] \leq 2^t \cdot \mathbb{E} \left[ \sup_{X \in \mathcal{X}} \left| \sum_{i=1}^d \varepsilon_i \cdot \left( \mathbb{I}_{\{N_Y,i=0\}} \cdot \log \frac{F(\infty, X_i)}{F(\infty, 0)} + \sum_{s=1}^n \mathbb{I}_{\{N_Y,i=s\}} \cdot \log \frac{p(Y_{i,s} - X_{i,s})}{p(Y_{i,s})} \right) \right|^t \right],
\]

(23)

where $\{\varepsilon_i\}_{i \in [d]}$ are i.i.d. Rademacher random variables. Note that, for $x, \hat{x} \in \mathbb{R}$,

\[
\log p(Y_{i,s} - x) - \log p(Y_{i,s} - \hat{x}) \leq |x - \hat{x}| \cdot \sup_{|u| \leq \gamma} \frac{d(\log F(\infty, u))}{du} \leq |x - \hat{x}| \cdot L_\gamma(p),
\]

and

\[
|\mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)]|^t \leq |x - \hat{x}| \cdot \sup_{|u| \leq \gamma} \frac{d(\log \int_{-\infty}^{\infty} p(y)dy)}{du} \leq |x - \hat{x}| \cdot L_\gamma(p).
\]

At this point, we can combine the contraction principle (cf. Theorem 2) with (22) to obtain the following.

\[
\mathbb{E} \left[ \sup_{X \in \mathcal{X}} |\mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)]|^t \right] \leq 2^t \cdot 2^t \cdot \mathbb{E} \left[ (L_\gamma(p))^t \cdot \sup_{X \in \mathcal{X}} \left| \sum_{i=1}^d \varepsilon_i \cdot \left( \mathbb{I}_{\{N_Y,i=0\}} \cdot X_i^\ast + \sum_{s=1}^n \mathbb{I}_{\{N_Y,i=s\}} \cdot X_{i,s} \right) \right|^t \right] \leq 4^t \cdot \mathbb{E} \left[ (L_\gamma(p))^t \cdot \sup_{X \in \mathcal{X}} \left( \sum_{i=1}^d \varepsilon_i^2 \right)^{t/2} \left( \sum_{i=1}^d \sum_{s=1}^n \mathbb{I}_{\{N_Y,i=s\}} \cdot X_{i,s} \right)^{t/2} \right]
\]

\[
\leq 4^t \cdot \mathbb{E} \left[ (L_\gamma(p))^t \cdot \sup_{X \in \mathcal{X}} \left( \sum_{i=1}^d \varepsilon_i^2 \right)^{t/2} \left( \sum_{i=1}^d \sum_{s=1}^n \mathbb{I}_{\{N_Y,i=s\}} \cdot X_{i,s} \right)^{t/2} \right]
\]

\[
\leq 4^t \cdot \mathbb{E} \left[ (L_\gamma(p))^t \cdot \sup_{X \in \mathcal{X}} \left( \sum_{i=1}^d \varepsilon_i^2 \right)^{t/2} \left( \sum_{i=1}^d \sum_{s=1}^n \mathbb{I}_{\{N_Y,i=s\}} \cdot X_{i,s} \right)^{t/2} \right] = (4L_\gamma(p) \cdot d_\gamma)^t,
\]

where (i) and (ii) follow from the Cauchy-Schwartz inequality and the fact that, for $X \in \mathcal{X}$, $\|X\|_\infty \leq \gamma$, respectively. Now using Markov’s inequality, it follows from (24) that

\[
\mathbb{P} \left( \sup_{X \in \mathcal{X}} |\mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)]| \geq C_0 L_\gamma(p) \cdot \gamma d \right) \leq \mathbb{E} \left[ \sup_{X \in \mathcal{X}} |\mathcal{Z}_Y(X) - E [\mathcal{Z}_Y(X)]|^t \right] \leq \left( \frac{4}{C_0} \right)^t \leq \frac{1}{d + n},
\]

where (i) follows from (24); and (ii) follows by setting $C_0 \geq \frac{1}{d}$ and $t = \log(d + n)$.

**C. Recovering the network parameters**

Let’s denote the recovered matrix $\hat{M}$ as $\hat{M} = M + E$, where $E$ is the perturbation matrix that has bounded Frobenius norm (cf. (19)). Now the task of recovering the parameters of the single-layer ReLU-network is equivalent to solving for $A$ given $\hat{M} = M + E = AC + E$. In our setting where we have $A \in \mathbb{R}^{d \times k}$ and $C \in \mathbb{R}^{k \times n}$ with $d > k$ and $n > k$, $M$ is a low-rank matrix with its column space spanned by the columns of $A$. Therefore, as long as the generative model ensures that the matrix $M$ has its singular values sufficiently bounded away from $0$, we can resort to standard results from matrix-perturbation theory and output top $k$ left singular vectors of $\hat{M}$ as a candidate for the orthonormal basis for the column space of $M$ or $A$. Note that, even without the perturbation $E$ we could only hope to recover the column space of $A$ and not the exact matrix $A$. Let $U_k$ and $\hat{U}_k$ be the top $k$ left singular vectors of $M$ and $\hat{M}$, respectively. Let $\sigma_k$, the smallest non-zero singular value of $M$, is at least $\delta > 0$. Then,
it follows from [17] that there exists an orthogonal matrix
$O \in \mathbb{R}^{k \times k}$ such that
\[
\|U_k - \hat{U}_k O\|_F \leq \frac{2^{3/2}(2\sigma_1 + \|E\|) \cdot \min\{\sqrt{k}\|E\|, \|E\|_F\}}{\delta^2}.
\]
which is a guarantee that the column space of $U_k$ is preserved within an error of $O(d)$ in Frobenius norm by the column space of $\hat{U}_k$. Note that $\sigma_1$ is the largest singular value of $M$.

IV. ROBUST RECOVERY IN SINGLE-LAYER RELU-NETWORK

We now explore the second basic question that arises in the context of reconstructing a signal vector belonging to the underlying generative model from its noisy version. We are given a vector $v \in \mathbb{R}^d$, which is obtained by adding noise to a valid signal vector $y \in \mathbb{R}^d$ that is well modeled by a single-layer ReLU-network, i.e.,
\[
v = y + e + w = \text{ReLU}(Ac + b) + e + w.
\]
Here, $w$ denotes the (dense) noise with bounded norm and $e$ contains (potentially) large corruptions (a.k.a. outliers). We assume the number of outliers $\|e\|_0$ is at most $s$. The robust recovery problem in ReLU-networks corresponds to obtaining an estimate $\hat{c}$ of the true latent vector $c$ from the corrupt observation vector $v$ such that the distance between $\hat{c}$ and $c$ is small. A related problem of denoising only focuses on obtaining an estimate $\hat{y}$ which is close to $y$ (cf. (5)). In what follows, we focus on the setting where the weight matrix $A$ is a random matrix with i.i.d. standard Gaussian entries.

Furthermore, we assume that the outlier vector is oblivious in its nature, i.e., the error vector is not picked in an adversarial manner given the knowledge of $A$ and $c$.

Note that [1] studies a problem which is equivalent to recovering the latent vector $c$ from the observation vector generated by a single-layer ReLU-network without the presence of outliers. In that sense, our work is a natural generalization of [1] and presents a recovery method which is robust to outliers as well. However, our approach significantly differs from that in [1], where the author analyze the convergence of the gradient descent method to the true representation vector $c$. In contrast, we rely on the recent work of [22] to employ the LASSO method to recover the representation vector $c$ (and the outlier vector $e$). Note that, the work of [22] also does not consider outliers. After applying the key technique from [22], our main effort effort goes into dealing with the outliers.

Given $v = \text{ReLU}(Ac + b) + e + w$, which corresponds to the corrupted non-linear observations of $c$, we aim to fit a linear model to these observations by solving the following optimization problem.

\[
\min_{v \in \mathbb{R}^d, e \in \mathbb{R}^d} \frac{1}{2d} \|v - Ac - e\|_2^2 + \lambda \|e\|_1.
\]  

In this formulation, the $\ell_1$-regularizer is included to encourage the sparsity in the estimate for outlier vector. Since the number of observations is greater than the dimension of $c$, we don’t necessarily require $c$ to belong to a restricted set, as done in the robust LASSO methods for linear measurements (see e.g., [23]). The following result characterizes the performance of the program proposed in (26).

**Theorem 4.** Let $A \in \mathbb{R}^{d \times n}$ be a random matrix with i.i.d. standard Gaussian random variables as its entries and $v$ satisfies
\[
v = \text{ReLU}(Ac^* + b) + e^* + w,
\]
where $\|c^*\|_2 = 1$, $\|e^*\|_0 \leq s$ and $\|w\|_\infty \leq \delta$. Let $\mu$ be defined as $\mu = \mathbb{E}[\text{ReLU}(g + b) \cdot g]$, where $g$ is a standard Gaussian random variable and $b$ is a random variable that represents the bias in a coordinate in (27). Let $(\hat{e}, \hat{c})$ be the outcome of the program described in (26) with $\lambda \geq \log(d)/d$. Then, for a large enough absolute constant $\tilde{C}$ that depends on $\delta$, the following holds with high probability.

\[
\|v - \mu c^* - \hat{c}\|_2 + \|e^* - \hat{e}\|_2/\sqrt{d} \leq \tilde{C} \cdot \max \left\{ \sqrt{(k \log k)/d}, \sqrt{(s \log d)/d} \right\}.
\]  

(28)

The rest of this section is devoted to the proof of this theorem. Assume that
\[
\hat{c} = \mu c^* + h \text{ and } \hat{e} = e^* + f.
\]

Furthermore, for $c \in \mathbb{R}^d$ and $e \in \mathbb{R}^k$, we define
\[
\mathcal{L}(c, e) = \frac{1}{2d} \|y - Ac - e\|_2^2 + \lambda \|e\|_1.
\]

Let $S = \{i \in [d] : c_i^* \neq 0\}$ be the support of the vector $c^*$ such that $|S| = s$. Given a vector $a \in \mathbb{R}^d$ and a set $T \subseteq [d]$, we use $a_T$ to denote the vector obtained by restricting $a$ to the indices belonging to $T$. Note that
\[
\mathcal{L}(\hat{c}, \hat{e}) - \mathcal{L}(\mu c^*, e^*)
\]

\[
= \frac{1}{2d} \|v - \mu Ac^* - e^* - Ah - f\|_2^2 + \lambda \|e^* + f\|_1 - \frac{1}{2d} \|v - \mu Ac^* - e^*\|_2^2 - \lambda \|e^*\|_1
\]

\[
= \frac{1}{2d} \|Ah + f\|_2^2 + \lambda \|\langle (e^* + f^*)_{S^c} \rangle\|_1 + \|f_{S^c}\|_1 - \|e^*\|_1
\]

\[
\geq \frac{1}{2d} \|Ah + f\|_2^2 + \lambda \|\langle f_{S^c}\|_1 - \|f_S\|_1\| - \frac{1}{d} \|\text{ReLU}(Ac^* + b) - \mu Ac^* + w, Ah + f\|
\]  

(29)

(30)

(31)
where (i) and (ii) follow from (5) and the triangle inequality, respectively. Since \((\hat{c}, \hat{e})\) is solution to the program in (26), we have
\[
\mathcal{L}(\hat{c}, \hat{e}) - \mathcal{L}(\mu c^*, e^*) \leq 0. \tag{32}
\]
By combining this with (31), we obtain that
\[
\frac{1}{2d} \left\| Ah + f \right\|^2 \leq \frac{1}{d} \left( \text{ReLU}(Ac^* + b) - \mu Ac^* + w, Ah + f \right) + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) \tag{33}
\]
We now complete the proof in two steps where we obtain universal lower and upper bounds on the left hand side and the right hand side of (33), respectively, that hold with high probability.

**Upper bound on the RHS of (33).** Let’s define
\[
z = \text{ReLU}(Ac^* + b) - \mu Ac^*. \tag{34}
\]
Note that
\[
\frac{1}{d} \cdot \langle z + w, Ah + f \rangle + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) = \frac{1}{d} \cdot \langle z + w, Ah \rangle + \frac{1}{d} \cdot \langle z + w, f \rangle + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) \leq \frac{1}{d} \cdot \langle z + w, Ah \rangle + \frac{1}{d} \cdot \| z + w \|_\infty \| f \|_1 + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) = \frac{1}{d} \cdot \langle z + w, Ah \rangle + (\lambda - \frac{1}{d} \cdot \| z + w \|_\infty) \| f_{SC} \|_1
\]
where (i) follows from the Hölder’s inequality. We now employ [22, Lemma 4.3] to obtain that\(^3\)
\[
\sup_{h \in \mathbb{R}^k} \langle z, Ah \rangle \leq C (\sqrt{k} \sigma + \eta) \sqrt{d} \cdot \| h \|_2, \tag{36}
\]
where \(C\) is an absolute constant and
\[
\sigma^2 := \mathbb{E}[(\text{ReLU}(g + b) - \mu g)^2] \quad \eta^2 := \mathbb{E}[g^2 \cdot (\text{ReLU}(g + b) - \mu g)^2]
\]
with \(g\) being a standard Gaussian random variable. Now we can combine (35) and (36) to obtain the following.
\[
\frac{1}{d} \cdot \langle z + w, Ah + f \rangle + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) \leq C (\sqrt{k} \sigma + \eta) \sqrt{d} \cdot \| h \|_2 + \frac{\sqrt{k}}{d} \| A^T w \|_\infty \| h \|_2 + (\lambda - \frac{1}{d} \cdot \| z + w \|_\infty) \| f_{SC} \|_1 \tag{37}
\]
\[
\frac{1}{d} \cdot \langle z + w, Ah + f \rangle + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) \leq C (\sqrt{k} \sigma + \eta) \sqrt{d} \cdot \| h \|_2 + \frac{\sqrt{k}}{d} \| A^T w \|_\infty \| h \|_2 + \lambda (\| f_S \|_1 - \| f_{SC} \|_1)
\]

\(^3\)In [22], Plan and Vershynin obtain the bound in terms of the Gaussian width [29] of the cone which the vector \(h\) belongs to. However, in our setup where we do not impose any specific structure on \(c^*\), this quantity is simply \(O(\sqrt{k})\).

\[
\frac{1}{d} \cdot \langle z + w, Ah + f \rangle + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) \leq C (\sqrt{k} \sigma + \eta) \sqrt{d} \cdot \| h \|_2 + \frac{\sqrt{k}}{d} \| A^T w \|_\infty \| h \|_2 + 2 \lambda \sqrt{d} \| f \|_2, \tag{39}
\]
where (i) and (ii) follow by setting \(\lambda \geq 2 \| z + w \|_\infty / d\) and using the fact that \(\| f_S \|_1 \leq \sqrt{k} \| f_S \|_2 \leq \sqrt{k} \| f \|_2\). We can further simplify the bound in (39) as follows.
\[
\frac{1}{d} \cdot \langle z + w, Ah + f \rangle + \lambda (\| f_S \|_1 - \| f_{SC} \|_1) \leq \max \left\{ C \left( \frac{\sqrt{k} \sigma + \eta}{\sqrt{d}} \right) + \frac{\sqrt{k}}{d} \| A^T w \|_\infty, 2 \lambda \sqrt{d} \right\} \left( \| h \|_2 + \frac{\| f \|_2}{\sqrt{d}} \right). \tag{40}
\]

**Lower bound on the LHS of (33).** By combining (33) and (38), we get that
\[
\frac{1}{2d} \| Ah + f \|_2^2 \leq C (\sqrt{k} \sigma + \eta) \sqrt{d} \cdot \| h \|_2 + \frac{\sqrt{k}}{d} \| A^T w \|_\infty \| h \|_2 + \left( \frac{\sqrt{k}}{d} \right) \| f_{SC} \|_1 - \left( \frac{\| z + w \|_\infty}{d} \right) \| f_{SC} \|_1, \tag{41}
\]
Note that we have picked \(\lambda \geq 2 \| z + w \|_\infty / d\). Since the left hand side of (41) is non-negative, we find that the tuple \((h, f)\) belongs to the following restricted set.
\[
(h, f) \in \mathcal{R} := \left\{ h \in \mathbb{R}^k, f \in \mathbb{R}^d : \lambda \cdot \| f_{SC} \|_1 \leq 3 \lambda \| f_S \|_1 + C (\sqrt{k} \sigma + \eta) \sqrt{d} + \sqrt{k} \| A^T w \|_\infty / d \| h \|_2 \right\}. \tag{42}
\]
As a result, in order to lower bound (33), we lower bounding the following quantity for every \((h, f) \in \mathcal{R}\).
\[
\frac{1}{2d} \cdot \| Ah + f \|_2^2. \tag{43}
\]
Towards this, we employ Lemma 2 at the end of this section, which gives us that, for every \((h, f) \in \mathcal{R}\), with high probability, we have
\[
\frac{1}{2d} \cdot \| Ah + f \|_2^2 \geq \frac{1}{128} \left( \| h \|_2 + \frac{\| f \|_2}{\sqrt{d}} \right)^2. \tag{44}
\]
**Completing the proof.** It follows from (33), (40), and (44) that
\[
\frac{1}{128} \left( \| h \|_2 + \frac{\| f \|_2}{\sqrt{d}} \right)^2 \leq \max \left\{ C \left( \frac{\sqrt{k} \sigma + \eta}{\sqrt{d}} \right) + \frac{\sqrt{k}}{d} \| A^T w \|_\infty, 2 \lambda \sqrt{d} \right\} \left( \| h \|_2 + \frac{\| f \|_2}{\sqrt{d}} \right)
\]
or
\[
\| h \|_2 + \frac{\| f \|_2}{\sqrt{d}} \leq 128 \max \left\{ C \left( \frac{\sqrt{k} \sigma + \eta}{\sqrt{d}} \right) + \frac{\sqrt{k}}{d} \| A^T w \|_\infty, \right\}
\]
Now using the fact that $\|w\|_\infty \leq \delta$ and $A$ is an i.i.d. standard Gaussian matrix, we can obtain the following bound from (45), which holds with high probability.

$$\|h\|_2^2 + \frac{\|f\|_2^2}{\sqrt{d}} \leq \tilde{C} \max \left\{ \sqrt{\frac{k \log k}{d}}, \sqrt{s \log d} \right\},$$

(46)

where $\tilde{C}$ is a large enough absolute constant.

Here we state the special form of a result that was obtained in [23] for the generals setting, where one may potentially require the vector $c$ to be sparse as well. This result is utilized in the proof of Theorem 4 above. We refer the reader to [28, Appendix C] for a detailed proof of the result.

**Lemma 2.** Let $A \in \mathbb{R}^{d \times n}$ be random matrix that has i.i.d. standard Gaussian entries. Furthermore, let $R \subset \mathbb{R}^d \times \mathbb{R}^d$ be as defined in (42). Then, with probability at least $1 - e^{\exp(-\tilde{c}d)}$, we have

$$\frac{1}{2d} \cdot \|Ah + f\|_2 \geq \frac{1}{128} \left( \|h\| + \frac{\|f\|}{\sqrt{d}} \right)^2 \quad \forall (h, f) \in R.$$

(47)

Here, $c, \tilde{c} > 0$ are absolute constants.

**V. Conclusion**

For ReLU activation networks, we have studied two basic problems of any generative model: network parameter learning and signal recovery from noisy observations. We provided provable guarantees for both the problems for a single-layer network. Although the network parameter learning problem has been a subject of intense study for the last few years, to the best of our knowledge theoretical guarantees for this were never provided even for single layer networks. Previous results for the recovery problem existed, and our contribution is to obtain guarantees even in the presence of large but sparse noise (outliers). While for the recovery problem our method extends to multiple layers, extension of network parameter learning to multiple layers does not appear to be straightforward. We view such extensions to be valuable potential follow-up work.

**References**


