EXACTLY DECODING A VECTOR THROUGH RELU ACTIVATION

Samet Oymak and M. Salman Asif

Department of Electrical and Computer Engineering
University of California, Riverside

ABSTRACT

We consider learning a d-dimensional parameter $\boldsymbol{w}$ through nonlinear input/output relation governed by ReLU activation. We study a supervised learning setup in which we want to decode $\boldsymbol{w}$ from input/output pairs $(\mathbf{x}, \boldsymbol{y})$. We consider an additive model with nonlinear ReLU activation that can be represented as $\boldsymbol{y} = \sum_{k=1}^{d} \mathrm{ReLU}(\boldsymbol{w}[k] + \mathbf{x}[k])$. Such a model appears in representation learning and recommendation systems where $\boldsymbol{w}$ corresponds to an unknown embedding of a user or item and the $\mathbf{x}$ correspond to embedding of known probe vectors. In this paper, we show that a gradient descent algorithm linearly converges with $\mathcal{O}(d)$ samples and quickly finds the true parameter $\boldsymbol{w}$ under mild assumptions. Our assumptions are in terms of the input distribution that captures the fundamentals of the problems. We also demonstrate the performance of our algorithm with numerical simulations.

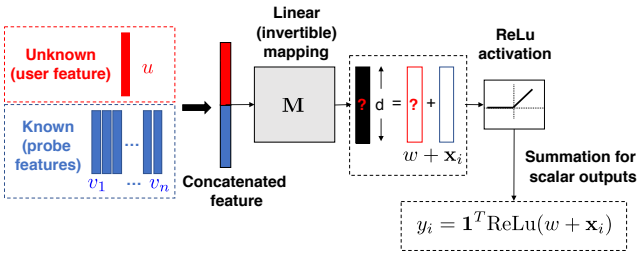


Fig. 1: Block diagram of the encoding model in which an unknown feature vector is mapped to a scalar output of a single-layer neural network. Assume that the unknown feature vector is combined with a known probe signal using an invertible linear function. The output of the linear function can be represented as $\boldsymbol{w} + \mathbf{x}_i$, where $\boldsymbol{w}$ is an unknown signal that depends on the hidden feature vector and $\mathbf{x}_i$ is a known signal that depends on the probe signals. $\boldsymbol{w} + \mathbf{x}_i$ passes through a ReLU activation function, and we observe sum of all the positive entries in $\boldsymbol{w} + \mathbf{x}_i$ as $y_i = \mathbf{1}^T \max(0, \boldsymbol{w} + \mathbf{x}_i)$. Our goal is to recover $\boldsymbol{w}$ from minimum possible samples in a computationally efficient manner. Our main result shows that with $\mathcal{O}(d)$ samples and using a centered gradient descent method, we can linearly converge to the true $\boldsymbol{w}$.

1. INTRODUCTION

In this paper we study a problem of recovering a d-dimensional parameter from its nonlinear measurements. We consider a single layer neural network that maps vector inputs to scalar outputs by first applying a linear weight matrix to each input, followed by a rectified linear unit (ReLU) activation and a summation (pooling operation). In particular, we want to recover a signal $\boldsymbol{w}^\star$ from input/output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where each y_i can be represented as a function of $\mathbf{x}_i + \boldsymbol{w}^\star$. Such additive models often appear in representation learning and recommendation systems [1–4], where $\boldsymbol{w}^\star$ represents a hidden information about some item or user and $\mathbf{x}_i$ represent features related to users actions (e.g. features of movies watches) that can help infer $\boldsymbol{w}^\star$. In practice, denoting user and video representation by $\boldsymbol{u}$ and $\boldsymbol{v}_i$, and the input layer of the neural recommendation model by $\boldsymbol{M} = [\boldsymbol{M}_u \ \boldsymbol{M}_v]$, $\boldsymbol{w}^\star$ and $\mathbf{x}_i$ corresponds to the features at the end of the first layer obtained via $\boldsymbol{w}^\star + \mathbf{x}_i = [\boldsymbol{M}_u \ \boldsymbol{M}_v] \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v}_i \end{bmatrix}$. Later, these features are passed through activations and other layers. An illustration of the encoding model is depicted in Fig. 1.

This paper studies the additive learning problem (1) and provides non-asymptotic algorithmic guarantees for ReLU activation. We show that, nonlinearities actually facilitate learning and gradient descent quickly and globally converges to the ground truth representation $\boldsymbol{w}^\star$ is possible. This is in contrast with related works on shallow fully-connected neural networks where gradient descent can get stuck on local minima and convergence rate suffers from growing number of hidden units even with good initialization [5, 6]. Furthermore, the sample complexity of gradient descent is near minimal, namely $n \geq \mathcal{O}(d)$.

Our second contribution is characterizing the problem parameters that enables convergence. The existing literature mostly focuses on Gaussian or separable data [5, 7, 8]. We show convergence results for arbitrary distributions (e.g. heavy tailed). In particular, convergence properties are in terms of the tail probabilities.

Finally, we introduce a variation of gradient descent algorithm that drastically speeds up the convergence compared to vanilla GD. This speed up occurs due to a centering trick that removes the impact of expectation from the system. This strategy was inspired from the label debiasing argument utilized in the recent work [8]. It can also be seen in similar spirit to techniques such as batch normalization that attempts to whiten the features. We demonstrate the benefit of this numerically.

1.1. Related Work

There is a growing literature on the theoretical aspects of deep learning and provable algorithms for training neural networks. Most of the existing results are concerned with feedforward networks [5, 9–17]. [9, 14–16] consider learning fully-connected shallow networks with gradient descent. [8, 10, 13] address convolutional neural networks; which is an efficient weight-sharing architecture. [7, 18] studies over-parameterized networks when data is linearly separable. [8, 14] utilize tensor decomposition techniques for learning feedforward neural nets. [11, 19] study signal recovery from ReLU activation for related linear models.

2. PROBLEM STATEMENT

Let ϕ be the ReLU nonlinearity $\phi(a) = \max\{0, a\}$. Given a latent parameter $\boldsymbol{w}^\star \in \mathbb{R}^d$ and input data $\mathbf{x} \in \mathbb{R}^d$, we define the following additive nonlinearity f defined as

$$f(\mathbf{x}; \boldsymbol{w}^\star) = \sum_{k=1}^{d} \phi(\boldsymbol{w}^\star[k] + \mathbf{x}[k]) \equiv \mathbf{1}^T \phi(\boldsymbol{w}^\star + \mathbf{x}), \qquad (1)$$

where $\boldsymbol{w}^\star[k], \mathbf{x}[k]$ represent kth entry in vectors $\boldsymbol{w}^\star, \mathbf{x}$. Note that output is obtained by entrywise interactions with input data and ReLU activation. These interactions are entangled as we sum over them. Our goal will be understanding the optimization landscape of this function; which is connected to representation learning. To do this, we consider a dataset of n samples $(\mathbf{x}_i, y_i)$ and assume that, the latent parameter $\boldsymbol{w}^\star$ is responsible for the labels i.e. the labels satisfy $y_i = f(\mathbf{x}_i; \boldsymbol{w}^\star)$ for all $1 \le i \le n$.

To learn latent parameter $\boldsymbol{w}^\star$, we will study a quadratic loss function over our dataset and consider empirical risk minimization problem. Given a vector $\boldsymbol{w} \in \mathbb{R}^d$, this loss is given by

$$\mathcal{L}(\boldsymbol{w}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \boldsymbol{w}))^2. \qquad (2)$$

This loss will be minimized using first order methods, such as, gradient descent (GD). The next section describes our approach to this problem.

2.1. Algorithm

Our algorithm is a novel variant of GD that includes projection and centering. Typically, we can run the GD iterations

$$\hat{\boldsymbol{w}} = \boldsymbol{w} - \mu \nabla \mathcal{L}(\boldsymbol{w}). \qquad (3)$$

Empirically, as discussed in Section 4, we observed that GD does not converge quickly. To address this, we apply two modifications. First observe that labels have positive mean because ReLU output is always positive. In practice, this bias slows down the convergence rate. To speed it up, we study the centered gradient descent where we center residuals with the empirical average $\bar{r}(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i; \boldsymbol{w}) - y_i)$. With this, the centered gradient for our loss is given by

$$\nabla \hat{\mathcal{L}}(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i; \boldsymbol{w}) - y_i - \bar{r}(\boldsymbol{w})) \mathbf{1}(\boldsymbol{w} + \mathbf{x}_i). \qquad (4)$$

Here, $\mathbf{1}$ is the entrywise step function that corresponds to ReLU derivative. $\mathbf{1}(x) = 1$ if $x \ge 0$ and 0 if $x < 0$. Secondly, we assume entries of $\boldsymbol{w}^\star$ are bounded by a quantity Θ. This assumption is necessary for learning if data has bounded entries as well. We utilize this information by projecting our solution to the set $\mathcal{C}_\Theta = \{\boldsymbol{w} \in \mathbb{R}^d \mid \|\boldsymbol{w}\|_\infty \le \Theta\}$. Combining these two modifications, given current estimate $\boldsymbol{w}$, the next iterate of our algorithm is given by

$$\hat{\boldsymbol{w}} = \mathcal{P}_{\mathcal{C}_\Theta}(\boldsymbol{w} - \mu \nabla \hat{\mathcal{L}}(\boldsymbol{w})),$$

where $\mathcal{P}_{\mathcal{C}_\Theta}$ is the projection operator that clips the entries of a vector that are outside of $\pm\Theta$ interval. To state our technical result, we need the following quantities that capture the distribution of the data.

Definition 2.1 (Critical quantities) *Let X be a random variable with finite first moment (i.e. $\mathbb{E}|X| < \infty$) and let $\theta > 0$ be a scalar. Define θ-tail functions $\mathcal{T}$ and $\mathcal{S}$ as*

$$\mathcal{T}(\theta) = \mathbb{P}(X \ge \theta)\mathbb{P}(X < -\theta),$$
$$\mathcal{S}(\theta) = \sup_{|\gamma| \le \theta} \mathbb{P}(X \ge \gamma)\mathbb{P}(X < -\gamma).$$

Observe that $\mathcal{T}(\theta) \le \mathcal{S}(\theta) \le 1/4$.

Observe that, if X has a continuous distribution, as Θ gets smaller, both of these quantities converge to $\mathbb{P}(X \ge 0)\mathbb{P}(X < 0)$. If the probability density function f_X is bounded, (i.e. satisfies $f_X(t) \le B$), we have that $\mathcal{T}(\theta) \ge \mathcal{T}(0) - \Theta B$. .We also always have $\mathcal{T}(\Theta) \le \mathcal{S}(\Theta)$ which can be upper bounded in a similar fashion. In summary, these are fairly manageable quantities which will govern our main result.

3. MAIN RESULT

The following theorem is our main result on iterative convergence of the proposed algorithm.

Theorem 3.1 *Suppose X is a random variable with finite first moment. Let $(\mathbf{x}_i)_{i=1}^n$ be independent vectors with i.i.d. entries distributed with X. Let K be an integer and suppose $n = K\bar{n}$ where $\bar{n} \ge \mathcal{O}(d \log n / \mathcal{T}(\Theta)^2)$. Suppose labels satisfy $y_i = f(\mathbf{x}_i; \boldsymbol{w}^\star)$. Pick a learning rate $\mu = \frac{2\mathcal{T}(\Theta)}{9\mathcal{S}(\Theta)^2}$. Split n datapoints into K batches of equal size. Let $\mathcal{L}_i$ be the loss function corresponding to the ith batch. Starting from an initial point $\boldsymbol{w}_0$, consider the iterations*

$$\boldsymbol{w}_{\tau+1} = \mathcal{P}_{\mathcal{C}_\Theta}(\boldsymbol{w}_\tau - \mu \nabla \hat{\mathcal{L}}_\tau(\boldsymbol{w}_\tau)).$$

With probability $1 - 2dK\bar{n}^{-10}$, for all $K \ge \tau \ge 0$,

$$\|\boldsymbol{w}_\tau - \boldsymbol{w}^\star\|_{\ell_2}^2 \le \left(1 - \left(\frac{\mathcal{T}(\Theta)}{3\mathcal{S}(\Theta)}\right)^2\right)^\tau \|\boldsymbol{w}_0 - \boldsymbol{w}^\star\|_{\ell_2}^2.$$

This result implies that, as long as the data distribution X is rich enough (i.e. large tails), gradient descent can exactly decode the latent vector with linear rate of convergence. Unlike existing literature; which mostly focuses on normal distribution, our results apply to any distribution. In fact, perhaps counterintuitively, as the tail of the distribution becomes larger, convergence is faster! Also, note that, for constant K, our theorem needs only $\mathcal{O}(d \log n)$ samples; which is proportional to the parameter dimension, hence we achieve near-optimal sample complexity.

To prove this result, we first characterize the improvement due to a single data batch i.e. a single gradient iteration. We then apply this result K times and use a union bound to conclude. We use mini-batches due to technicalities (i.e. dependence of the iterates); however since we achieve linear rate of convergence, we need only $K = \mathcal{O}(\log(1/\varepsilon))$ to achieve ε accuracy. We also remark that setting $\mathcal{S}(\Theta) \le 1/4$, the convergence rate can be simplified to $1 - 16\mathcal{T}(\Theta)^2/9$. In the next section, we discuss the critical ideas in our proof strategy.

3.1. Approach

A standard idea for gradient descent analysis is ensuring gradient has a large component in the direction of the parameter error $\boldsymbol{w} - \boldsymbol{w}^\star$ i.e. strong convexity. The following lemma highlights this.

Lemma 3.2 *Given vectors $\boldsymbol{w}, \boldsymbol{w}^\star \in \mathcal{C}_\Theta$ and scalars $\beta \ge \alpha > 0$, suppose*

- $(\boldsymbol{w} - \boldsymbol{w}^\star)^T \nabla \hat{\mathcal{L}}(\boldsymbol{w}) \ge \alpha \|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2}^2$
- $\|\nabla \hat{\mathcal{L}}(\boldsymbol{w})\|_2 \le \beta \|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2}$.

$\hat{\boldsymbol{w}} = \mathcal{P}_{\mathcal{C}_\Theta}(\boldsymbol{w} - \mu \nabla \hat{\mathcal{L}}(\boldsymbol{w}))$. Then, setting learning rate $\mu = \alpha/\beta^2$, $\hat{\boldsymbol{w}}$ obeys $\|\hat{\boldsymbol{w}} - \boldsymbol{w}^\star\|_{\ell_2} \le \|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2}^2 (1 - \alpha^2/\beta^2)$.

The centering trick plays a critical role in our fast convergence result. Given a random variable X, define debiasing operator as $\mathbf{zm}(X) = X - \mathbb{E}[X]$. Suppose $(\mathbf{x}_i, y_i) \sim (\mathbf{x}, y)$ are i.i.d. samples where $y = f(\mathbf{x}; \boldsymbol{w}^\star)$. Observe that $\mathbb{E}[\bar{r}(\boldsymbol{w})] = \mathbb{E}[f(\mathbf{x}; \boldsymbol{w}) - y]$ hence, we have that

$$\begin{aligned}\mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})] &= \mathbb{E}[(f(\mathbf{x}; \boldsymbol{w}) - \boldsymbol{y} - \bar{r}(\boldsymbol{w}))\mathbf{1}(\boldsymbol{w} + \mathbf{x})] \\ &= \mathbb{E}[\mathbf{zm}(f(\mathbf{x}; \boldsymbol{w}) - \boldsymbol{y})\mathbf{1}(\boldsymbol{w} + \mathbf{x})] - \mathbb{E}[\mathbf{zm}(\bar{r}(\boldsymbol{w}))\mathbf{1}(\boldsymbol{w} + \mathbf{x})] \\ &= (1 - 1/n)\, \mathbb{E}[\mathbf{zm}(f(\mathbf{x}; \boldsymbol{w}) - \boldsymbol{y})\mathbf{1}(\boldsymbol{w} + \mathbf{x})]. \end{aligned} \tag{5}$$

With this, we decompose the centered gradient into an expectation term and two additional zero-mean terms via $\nabla \hat{\mathcal{L}}(\boldsymbol{w}) = \mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})] + n^{-1}(\nabla \mathcal{L}_1(\boldsymbol{w}) + \nabla \mathcal{L}_2(\boldsymbol{w}))$. The latter terms are

$$\nabla \mathcal{L}_1(\boldsymbol{w}) = \sum_{i=1}^n \mathbf{zm}(\mathbf{zm}(f(\mathbf{x}_i; \boldsymbol{w}) - y_i)\mathbf{1}(\boldsymbol{w} + \mathbf{x}_i)) \tag{6}$$

$$\nabla \mathcal{L}_2(\boldsymbol{w}) = \mathbf{zm}(\mathbf{zm}(\bar{r}(\boldsymbol{w})) \sum_{i=1}^n \mathbf{1}(\boldsymbol{w} + \mathbf{x}_i)) \tag{7}$$

The challenge in the analysis is two-folds. First, we need to understand, how beneficial is the expectation. This characterizes the convergence rate if we had access to infinite amount of data. Secondly, we shall view $\nabla \mathcal{L}_1(\boldsymbol{w}), \nabla \mathcal{L}_2(\boldsymbol{w})$ as noise and argue that, for sufficiently large n (i.e. $n \gtrsim d \log n$), their affect on the convergence will be small with high probability. This two-fold analysis is typical in empirical risk minimization problems [11, 12, 20]. In our case, it is rather straightforward to argue that $\nabla \mathcal{L}_2(\boldsymbol{w})$ term is small.

Population landscape: The next lemma is one of our key observations to analyze $\mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})]$ term.

Lemma 3.3 (Significant correlation) *Let X be a random variable with $\mathbb{E}|X| < \infty$. Given $a, b \in \mathbb{R}$ satisfying $\Theta > |a|, |b| \ge 0$. Set $\Phi_X(a, b) = \mathbb{E}_X[\mathbf{zm}(\phi(a + X) - \phi(b + X))\mathbf{1}(a + X)]$. We have that*

$$|a - b|\mathcal{S}(\Theta) \ge sgn(a - b)\Phi_X(a, b) \ge |a - b|\mathcal{T}(\Theta).$$

This shows that, if the randomness (i.e. diversity) in data is strong enough, $\phi(a + X) - \phi(b + X)$ will behave like $a - b$ in average, up to some constants $\mathcal{S}(\Theta), \mathcal{T}(\Theta)$. Recalling the form of the gradient (5) and applying Lemma 3.3 entrywise on $\mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})]$, we show below that gradient and residual $\boldsymbol{w} - \boldsymbol{w}^\star$ is highly correlated in expectation. captures the impact of the expectation term $\mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})]$

Corollary 3.4 (Expected strong convexity) *Suppose $\boldsymbol{w}, \boldsymbol{w}^\star \in \mathcal{C}_\Theta$. Generate independent vectors $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^d$ with i.i.d. entries distributed as random variable X. Then*

$$(\boldsymbol{w} - \boldsymbol{w}^\star)^T \mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})] \ge (1 - 1/n)\|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2}^2 \mathcal{T}(\Theta).$$

Furthermore, $\|\mathbb{E}[\nabla \hat{\mathcal{L}}(\boldsymbol{w})]\|_{\ell_2} \le (1 - 1/n)\mathcal{S}(\Theta)\|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2}$.

Finite sample analysis: Results so far, guarantees fast convergence with infinite data, by applying Lemma 3.2. The remaining task is obtaining finite sample complexity, which will be achieved by proving $\nabla \mathcal{L}_1(\boldsymbol{w})$ is small. This is achieved by utilizing ideas from non-asymptotic statistics literature [21]. The main idea is that $\nabla \mathcal{L}_1(\boldsymbol{w})$ is sum of n i.i.d. terms; hence, the variance of its entries should grow as n and its ℓ_2 norm should grow as $\sqrt{nd}$. To obtain such bounds in probability, we prove that each summand is a subexponential vector (see Def. 5.13 of [21]) and obtain the following ℓ_2 concentration bound.

Lemma 3.5 (Subexponential gradient) *Consider the setup in Lemma 3.2 Define per-sample gradient noise $h(\mathbf{x})$ as*

$$h(\mathbf{x}) = \mathbf{zm}(\mathbf{zm}(f(\mathbf{x}; \boldsymbol{w}) - f(\mathbf{x}; \boldsymbol{w}^\star))\mathbf{1}(\boldsymbol{w} + \mathbf{x})).$$

Then subexponential norm of $h(\mathbf{x})$ is at most $\mathcal{O}(\|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2})$. Furthermore, with probability $1 - \exp(-cd)$, the total noise $\nabla \mathcal{L}_1(\boldsymbol{w}) = \sum_{i=1}^n \mathbf{zm}(h(\mathbf{x}_i))$ satisfies

$$\|h\|_{\ell_2} \le \mathcal{O}(\|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2} \sqrt{dn}).$$

Finally, since $\nabla \hat{\mathcal{L}}(\boldsymbol{w})$ contains $n^{-1} \nabla \mathcal{L}_1(\boldsymbol{w})$, it means the noise we suffer grows as $\sqrt{d/n}\|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\ell_2}$ i.e. and no longer hurts the convergence rate in the regime $n \gtrsim d$!

4. NUMERICAL EXPERIMENTS

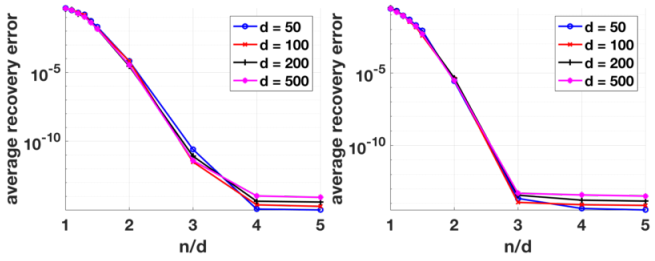
To demonstrate the empirical performance of our proposed method, we performed a number of numerical simulations. In all our experiments, we selected the true parameter $\boldsymbol{w}^\star \in \mathbb{R}^d$ as a random vector whose entries are independently and uniformly in interval $[-1, 1]$. We selected each $\mathbf{x}_i \in \mathbb{R}^d$ as a vector whose entries follow i.i.d. Gaussian distribution with zero mean and variance σ^2. We evaluated the performance of our recovery algorithm in terms of the values of the loss function in (2) and the relative recovery error after a fixed number of iterations. We define the relative recovery error as

$$\text{relative recovery error} = \frac{\|\hat{\boldsymbol{w}} - \boldsymbol{w}^\star\|_2^2}{\|\boldsymbol{w}^\star\|_2^2}, \tag{8}$$

where $\hat{\boldsymbol{w}}$ denotes the reconstructed values of the parameters. Our main simulation results are presented in Fig. 2 and Fig. 3.

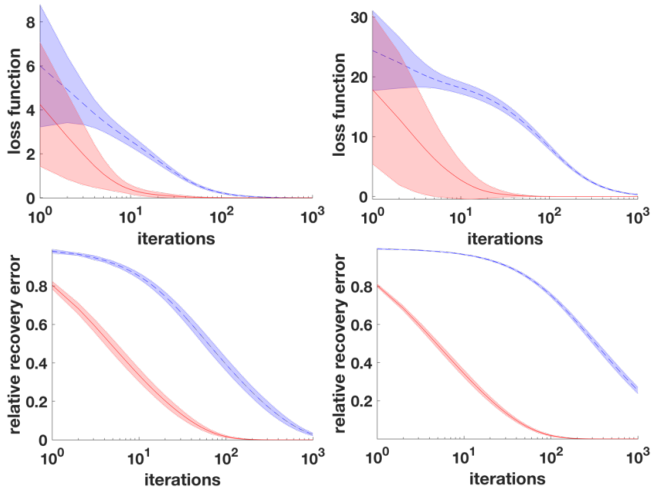
In our first experiment, we examine the performance of our algorithm for different number of samples. The results are summarized in Fig. 2, where we performed recovery experiments for different values of d, n, and σ and plotted average of the relative recovery error over 10 independent trials of each experiment. We report results for $d = 50,\ 100,\ 200,\ 500$ over different values of n/d in Fig. 2. As we can observe in Fig. 2(a)–(b) that for $n/d \geq 2$, signal recovery is almost always perfect. This observation matches our analysis that suggests $n \gtrsim d$ samples are sufficient for signal recovery. We also observed that increasing the strength of $\mathbf{x}_i$ (by changing σ) does not affect the performance of the algorithm. In all these experiments, we fixed the maximum number of iterations to 1000.

In our second experiment, we compared the performance of centered gradient descent (described in (4)) with standard gradient descent for a fixed step size. The results are presented in Fig. 3, where we plot the values of loss function and relative recovery error at every iteration of both algorithms, averaged over 10 independent trials. The red (solid) curves represent results for centered gradient descent and blue (dashed) curves represent results for standard gradient descent. The shaded regions indicate standard deviation of the results over 10 trials. We present experiments for $d = 100, n = 200$ in Fig. 3(a) and $d = 500, n = 1000$ in Fig. 3(b). Note that the x-axis (number of iterations) is plotted on a log scale for presentation. We observe that centered gradient descent converges to the true solution in nearly 100 iterations for both cases, whereas standard gradient descent takes much longer to converge. In all our experiments, we selected maximum possible step size (learning rate) μ for standard gradient descent, and multiplied μ by a large constant to select the step size for the centered gradient descent. Both algorithms use a fixed step size at every iteration. As noted in Sec. 2.1, the positive bias in standard gradient descent slows down its convergence rate, while the centered gradient descent does not suffer from the bias problem.



(a) $\sigma = 1$ **(b)** $\sigma = 10$

Fig. 2: Performance of recovery algorithm for different values of n/d. We showed that $n \gtrsim d$ samples would be sufficient for exact signal recovery. We observe that we get perfect recovery for $n/d > 2$. (a) and (b) compare the performance for different values of σ (strength of the probe signals), but the results look almost the same. Every point on these plots represents average relative recovery error over 10 independent trials.



(a) Loss function and reconstruction error for $d = 100, n = 200, \sigma = 5$

(b) Loss function and reconstruction error for $d = 500, n = 1000, \sigma = 5$

Fig. 3: Signal recovery error for $\sigma = 5$, bounded signal, and ReLU activation. The solid red curves plot average loss function while using mean-centered gradient descent as defined in (4). The dashed blue curves plot average loss function for standard gradient descent (without mean subtraction) with fixed step size. The shaded error bars in both plots indicate the standard deviation in the results over 10 trials. In all these experiments we selected the step size μ that provides fastest convergences for the standard gradient descent method. We provided a constant multiple of μ to our proposed gradient descent method. The x-axis shows number of iterations on a logarithmic scale.

5. CONCLUSIONS

We presented an algorithm for exact recovery of a vector from its nonlinear measurements through a ReLU activation function. We used an additive model for the measurements, which appears in representation learning and recommendation systems and showed that the unknown signal can be recovered exactly from $\mathcal{O}(d)$ samples. We also showed that by removing the mean of the residual from the gradient, we can minimize the loss function and converge to the true solution linearly.

6. REFERENCES

[1] Paul Covington, Jay Adams, and Emre Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 191–198.

[2] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[3] Shuai Zhang, Lina Yao, and Aixin Sun, "Deep learning based recommender system: A survey and new perspectives," *arXiv preprint arXiv:1707.07435*, 2017.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[5] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon, "Recovery guarantees for one-hidden-layer neural networks," *arXiv preprint arXiv:1706.03175*, 2017.

[6] Itay Safran and Ohad Shamir, "Spurious local minima are common in two-layer ReLU neural networks," *arXiv preprint arXiv:1712.08968*, 2017.

[7] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz, "Sgd learns over-parameterized networks that provably generalize on linearly separable data," *arXiv preprint arXiv:1710.10174*, 2017.

[8] Samet Oymak and Mahdi Soltanolkotabi, "End-to-end learning of a convolutional neural network via deep tensor decomposition," *arXiv preprint arXiv:1805.06523*, 2018.

[9] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *arXiv preprint arXiv:1707.04926*, 2017.

[10] Alon Brutzkus and Amir Globerson, "Globally optimal gradient descent for a convnet with gaussian inputs," *arXiv preprint arXiv:1702.07966*, 2017.

[11] Mahdi Soltanolkotabi, "Learning ReLUs via gradient descent," *arXiv preprint arXiv:1705.04591*, 2017.

[12] Samet Oymak, "Learning compact neural networks with regularization," *arXiv preprint arXiv:1802.01223*, 2018.

[13] Kai Zhong, Zhao Song, and Inderjit S Dhillon, "Learning non-overlapping convolutional neural networks with multiple kernels," *arXiv preprint arXiv:1711.03440*, 2017.

[14] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv preprint arXiv:1506.08473*, 2015.

[15] Yuanzhi Li and Yang Yuan, "Convergence analysis of two-layer neural networks with ReLU activation," in *Advances in Neural Information Processing Systems*, 2017, pp. 597–607.

[16] Song Mei, Andrea Montanari, and Phan-Minh Nguyen, "A mean field view of the landscape of two-layers neural networks," *arXiv preprint arXiv:1804.06561*, 2018.

[17] Samet Oymak, "Stochastic gradient descent learns state equations with nonlinear activations," *arXiv preprint arXiv:1809.03019*, 2018.

[18] Gang Wang, Georgios B Giannakis, and Jie Chen, "Learning ReLU networks on linearly separable data: Algorithm, optimality, and generalization," *arXiv preprint arXiv:1808.04685*, 2018.

[19] Arya Mazumdar and Ankit Singh Rawat, "Representation learning and recovery in the ReLU model," *arXiv preprint arXiv:1803.04304*, 2018.

[20] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[21] Roman Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.