Linearly Convergent Algorithms for Learning Shallow Residual Networks

Gauri Jagatap and Chinmay Hegde
Iowa State University
Ames, Iowa, USA
Email: {gauri,chinmay}@iastate.edu

***Abstract*—We propose and analyze algorithms for training ReLU networks with skipped connections. Skipped connections are the key feature of residual networks (or ResNets) which have been shown to provide superior performance in deep learning applications. We analyze two approaches for training such networks — gradient descent and alternating minimization — and compare convergence criteria of both methods. We show that under typical (Gaussianity) assumptions on the d-dimensional input data, both gradient descent and alternating minimization provably converge in a linearly convergent fashion, assuming any good enough initialization; moreover, we show that a simple "identity" initialization suffices. Furthermore, we provide statistical upper bounds which indicate that $n = \widetilde{O}(d^3)$ suffice to achieve this convergence rate. To our knowledge, these constitute the first global parameter recovery guarantees for shallow ResNet-type networks with ReLU activations.**

A full version of this paper is accessible at: https://gaurijagatap.github.io/assets/ISIT19.pdf

I. Introduction

Motivation. Deep neural networks have found considerable success in a wide range of machine learning applications. This has led to recent, renewed interest in analyzing (both new and old) algorithms for training such networks [1]–[3]. In particular, a class of neural networks called *residual networks* (or ResNets [4]) have emerged that show improved training performance when compared to standard dense architectures. These networks utilize so-called skipped connections, and several follow-up works such as DenseNets [5] and HighwayNets [6] have shown the merits of such architectures. Moreover, skipped connections lend to improved expressibility of neural networks [7] and empirically lead to a simpler optimization landscape with fewer local minima [8].

Our focus in this paper is on learning two-layer networks with rectified linear unit (ReLUs) activations with skipped connections. In general, two-layer ReLU networks are known to have spurious local minima [9], and hence convergence of first order methods to global optima such as (stochastic) gradient descent is not necessarily guaranteed. Moreover, due to the non-differentiability of ReLU activations, establishing linear convergence (also referred as geometric convergence) of gradient descent for shallow networks is challenging. Attempts to show convergence of gradient descent for shallow networks include strict assumptions on the smoothness of activation function [3] (which excludes ReLUs), restriction to analyzing population loss [10], [11], or requirement of strict orthogonality of weight vectors [10]. Moreover, many existing results for learning two-layer networks require computationally heavy tensor-based initializations [3], [12] and high sample complexity requirements [12], neither of which are practical.

Random initialization schemes pose their own set of challenges. Gradient descent can take exponential time to escape saddle points [13] and requires exponentially many epochs in terms of network depth under standard random initialization [14]. For two-layer networks, global linear convergence has only recently been established, albeit in the massively over-parametrized regime [15].

Our contributions. In this paper, we study two-layer ReLU networks with skipped connections, which are similar to those used in ResNet literature [1]. The contributions of this paper are two-fold.

First, we develop analytical tools to prove that such networks can be provably learned (in the sense of parameter estimation) with two algorithmic approaches: gradient descent (GD), and a new technique based on alternating minimization (AM). The building blocks of AM are reminiscent of that in gradient descent. However, AM is *parameter-free*: it does not involve any tuning parameters such as learning rate, damping factor and dropout ratio; moreover, the epoch complexity is also much lower. While GD is standard, the usage of alternating minimization in the context of training neural networks is non-standard, and to our knowledge, we are the first to propose such an approach.

Second, we utilize structural properties of residual networks to attest that an identity initialization is suitable for either of the above algorithms to exhibit *linear* convergence to the true parameters. Notably, we do not require any tensor based initialization schemes [3], [12], which are data dependent and instead rely on the specific structure of ResNet architectures.

Collectively, both the training algorithms can be supported via a rigorous analysis. Specifically, we use a generative modeling assumption where there is a "ground truth" (or teacher) network with k-hidden neurons and scalar output, that maps d-dimensional training samples x distributed according to a standard multivariate Gaussian distribution to labels y. We analyze the optimization of the empirical squared-error loss using both GD and AM under our identity initialization

[1] ResNets conventionally skip two layers of weights before identity mapping of the previous weights; in this paper we skip only one layer.

scheme and prove that $\tilde{O}(dk^2)$ are sufficient to achieve geometric convergence. Note that our analysis is much tighter, and our sample requirements are much lower, when compared to the only other known result in this regime $\tilde{O}(dk^9)$ from a recent paper [12]; however, our analysis makes the "fresh" resampling assumption in each iteration. Since we require exact parameter recovery, the results also generalize on test data drawn from the same distribution as training data.

Finally, we provide a range of numerical experiments that support our theoretical analysis. Our experiments show that both gradient descent and alternating minimization provide comparable performances; however alternating minimization has an improved epoch complexity and requires no cumbersome parameter tuning. We also compare our initialization against random initializations. Overall, our work can be viewed as a first step towards a new algorithmic approach and analysis for training ResNet-type ReLU networks.

Techniques. We first establish *local* linear convergence of both GD and AM for training two-layer ReLU networks. While (stochastic) gradient descent is popularly used in the training of neural networks, to our knowledge we are the first to introduce alternating minimization in this context.

At a high level, alternating minimization is based upon a simple (but key) idea that we call the "linearization" trick. Since ReLU networks simulate piecewise linear functions, if we *fix* the state of each hidden neuron ("on" or 1 if active, "off" or 0 if inactive), then the the mapping from x to the label y can be approximated using a *linear* neural network. We hence iteratively update the (i) "state" of ReLU neurons; hence locally linearizing and (ii) weight estimates by solving systems of linear equations [2] until convergence.

The conventional approach to showing linear convergence of gradient descent (or alternating minimization) relies on showing strong (marginal) convexity of the objective function in the neighborhood of global minimum [16]. In the case of ReLU networks, establishing this for empirical loss is not straightforward primarily due to the non-smoothness of the activation function. We circumvent this by borrowing key analytical tools from the recent literature on *phase retrieval* [17]–[19] which do not require any smoothness assumptions. This is also different from the uniform convergence criterion developed in [12] which requires a much higher sample complexity, and only guarantees convergence to a factor of statistical error that depends on the dimensionality of the network. On the contrary, with our analysis, we show convergence of both GD and AM to ϵ-small error in $log(1/\epsilon)$ iterations.

A key aspect in our analysis requires an independence assumption between subsequent iterates of the weights and the data matrix. In order to ensure this, a "re-sampling" approach is employed, which uses a fresh batch of samples in each iteration [3], [20], appending to the sample requirements by a multiplicative factor of $\log(1/\epsilon)$. This requirement is somewhat unsatisfactory, and one might be able to the leave-one-out trick introduced in [21] to circumvent this; we defer this to future work.

We leverage skipped connections to constrain the set of possible solutions $\mathbf{I} + W$, to a smaller subset of the space of optimization variables. We show that if the network weights are initialized to zeros (i.e. an identity mapping)[3] then both AM and GD converge *globally*.

Comparison with prior work. For random initializations, recent papers [10], [23] provide a symmetry-breaking convergence analysis for 2-layer ReLU networks where the weight vectors of the hidden layer possesses *disjoint supports*; moreover they only analyze population loss. Linear convergence guarantees for gradient descent for training two-layer networks with *smooth* activations were developed in [3], where they analyze empirical loss, along with a tensor initialization scheme to ensure global convergence requiring $\tilde{O}(dk^2)$ samples. For ReLU activation, a similar analysis is shown in [12], with uniform convergence arguments, but high sample requirements of $\tilde{O}(dk^9)$ (however they do not require resampling). Similarly, two-layer networks with skipped connections are studied in [11], however they only analyze population loss and hence no finite sample guarantees. Our convergence rate improves upon those in [11] and additionally we also provide finite sample guarantees for the same. A comparison of all algorithms and analysis is presented in Table I.

II. Mathematical model

a) Notation: Some of the notation used in the entirety of this paper are enlisted below. Scalars and vectors are denoted by small case letters, and matrices are denoted by upper case letters, with elements of both indexed by subscripts. The operation of ReLU is represented as $\sigma(\cdot)$. Indicator vector p stores the missing sign information from prior to the ReLU operation. The matrix $\mathbb{P} = \mathrm{diag}(p)$ represents the matrix with the elements of p along its diagonal and zero elsewhere. The symbol '$\circ$' denotes the element-wise product. Vectors and matrices with superscript '$*$' denote the ground truth. Vectorization (or flattening) of a matrix M is represented as $\mathrm{vec}(M)$. Small constants are represented by δ and large constants by C. The indicator function is denoted as $\mathbb{1}_{\{v\geq 0\}} := \frac{\mathrm{sign}(v)+\mathbf{1}}{2} = \sigma'(v)$, where each entry

$$\mathbb{1}_{\{v\geq 0\}_i} = \begin{cases} 1, & v_i \geq 0 \\ 0, & v_i < 0 \end{cases},$$

for all $i \in \{1 \dots n\}$, $\mathbf{I}$ is the identity matrix.

In each of the ReLU network architectures considered below, we assume that the training data consists of n i.i.d. samples $\{(x_1, y_1), \dots, (x_n, y_n)\}$, where each input $x_i \in \mathbb{R}^d$ is constructed from $\mathcal{N}(0, \mathbf{I})$, and the output y_i (scalar) obeys the generative model: $y_i = f^*(x_i)$ where f^* denotes the architecture. We denote the network weights by W^* and effective weights of the forward model are $W^* + \mathbf{I}$, due to

[2]These can be solved using LU/Cholesky decompositions or iterative conjugate gradient solvers, depending on the size of the problem.

[3]While preparing this paper we became aware of the work in [22], that also utilizes an identity initialization to train deep residual networks. However, their analysis only holds for *linear* neural networks.

TABLE I: *Comparison of algorithms and analysis for 1-hidden layer networks. Here, d, k, n denote dimensions of the data and hidden layer, number of samples, respectively. $\mathcal{O}_\epsilon(\cdot)$ hides polylogarithmic dependence on $\frac{1}{\epsilon}$. Alternating Minimization and (Stochastic) Gradient descent are denoted as AM and (S)GD respectively. "*" indicates re-sampling assumption.*

Alg.	Paper	Sample complexity	Convergence rate	Initialization	Type	Parameters
SGD	[11]	$\times$ (population loss)	$\mathcal{O}_\epsilon\left(\frac{1}{\epsilon}\right)$	Random	ReLU ResNets	step-size η
GD	[22]	$\times$ (population loss)	$\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$	Identity	Linear	step-size η
GD*	[3]	$\mathcal{O}_\epsilon\left(dk^2\cdot\text{poly}(\log d)\right)$	$\mathcal{O}_\epsilon\left(\log\frac{1}{\epsilon}\right)$	Tensor	Smooth (not ReLU)	step-size η
GD	[12]	$\mathcal{O}_\epsilon\left(dk^9\cdot\text{poly}(\log d)\right)$	$\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$	Tensor	ReLU	step-size η
GD*	(this paper)	$\mathcal{O}_\epsilon\left(dk^2\cdot\text{poly}(\log d)\right)$	$\mathcal{O}_\epsilon\left(\log\frac{1}{\epsilon}\right)$	Identity	ReLU ResNets	step-size η
AM*	(this paper)	$\mathcal{O}_\epsilon\left(dk^2\cdot\text{poly}(\log d)\right)$	$\mathcal{O}_\epsilon\left(\log\frac{1}{\epsilon}\right)$	Identity	ReLU ResNets	none

skipped connections. The number of hidden units k are equal to the input dimension d, due to skipped connections.

b) Two-layer ReLU networks: The problem of learning two-layer networks, with d-dimensional samples and k neurons in the hidden layer, can be expressed as a forward model: $f^*(X) = \sum_{q=1}^{k} v_q^*\sigma(Xw_q^*)$ where we have the weight matrix corresponding to the first layer $W^* := [w_1^* \dots w_q^* \dots w_k^*] \in \mathbb{R}^{d\times k}$, where each $w_q^* \in \mathbb{R}^d$, and we consider the second layer to be fixed, with entries $v_q^* \in \{+1, -1\}$ such that $v^* = [v_1^* \dots v_q^* \dots v_k^*]^\top \in \mathbb{R}^k$.

A special formulation of this problem is when there is a skipped connection between the network output and input. In such problems, the dimension at the input and output of the *residual block* is the same, and therefore $W^* \in \mathbb{R}^{d\times d}$ is a square matrix with $k = d$ columns. The effective mapping is

$$f_{res}^*(X) = \sum_{q=1}^{d} v_q^*\sigma(X(w_q^* + e_q)) = \sigma(X(W^* + \mathbf{I})v^*, \quad (1)$$

where we the weight matrix for the first layer $W^* \in \mathbb{R}^{d\times d}$ and e_q represents the q^{th} column of the d-dimensional identity matrix. In this paper, we focus on this formulation.

The learning problem essentially comprises of recovering the underlying mapping f_{res} (or f), from outputs y, such that $f_{res}(X)$ estimates $y = f_{res}^*(X)$. This can be formulated as a minimization of the following form,

$$\min_W \mathscr{L}(W) = \min_W \frac{1}{2n}\|f_{res}(X) - y\|_2^2 \quad (2)$$

where empirical risk $\mathscr{L}(W)$ is the ℓ_2-squared loss function.

III. Algorithm and Analysis

In this section, we establish two key algorithms, and lay groundwork for establishing their convergence.

First, we establish an observation. ReLU is a piece-wise linear transformation. In any iterative algorithm which estimates W^*, if W is the current weight estimate, and w_q represent the columns, one can represent a "linearized" mapping as follows. We introduce an new diagonal matrix $\mathbb{P}_q = \text{diag}(\mathbb{1}_{\{X(w_q+e_q)>0\}}), \forall q$ which stores the state of q^{th} hidden neuron for all samples. Then,

$$\begin{aligned} f_{res}(X) &= [v_1^*\mathbb{P}_1X \dots v_d^*\mathbb{P}_dX]_{n\times d^2}\cdot\text{vec}(W+\mathbf{I})_{d^2\times 1}, \\ &:= [A_1 \dots A_d]\cdot\text{vec}(W+\mathbf{I}) := B\cdot\text{vec}(W+\mathbf{I}). \end{aligned}$$

Note that the mapping is not truly linear, as B depends on W.

A. Gradient descent

The equation in (2) can be minimized via gradient descent. Since gradient descent is a first order approach, a careful initialization is desirable to ensure convergence to global minimum (which, in this case, is the teacher network parameters W^*). For the purpose of this description, we assume that an appropriate initialization W^0 exists, ensuring that $\|(W^0+\mathbf{I}) - (W^*+I)\|_F \le \delta\|W^*+I\|_F$. We discuss the initialization in further detail, in subsequent section III-D.

Now, the loss function can be written as:

$$\begin{aligned} \mathscr{L}(W) &= \frac{1}{2n}\|y - \sigma(X(W+\mathbf{I}))\cdot v^*\|_2^2 \\ &= \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \left(\sum_{q=1}^{d} v_q^*\sigma(x_i^\top(w_q+e_q))\right)\right)^2 \end{aligned}$$

Hence the gradient of loss function $\frac{\partial\mathscr{L}(W)}{\partial w_j}$ is

$$\begin{aligned} &\sum_{i=1}^{n}\left(-y_i + \sum_{q=1}^{d} v_q^*\sigma(x_i^\top(w_j+e_j))\right)\sigma'(x_i^\top(w_q+e_q))v_j^*x_i \\ &= -\frac{v_j^*}{n}X^\top\mathbb{P}_j(y - \sigma(X(W+\mathbf{I}))\cdot vv^*) \end{aligned}$$

Or alternatively,

$$\nabla\mathscr{L}(\text{vec}(W)) = -\frac{1}{n}B^\top(y - B\cdot\text{vec}(W+I)).$$

The gradient descent update rule is as follows:

$$\begin{aligned} \text{vec}(W^{t+1}) &= \text{vec}(W^t) - \eta\nabla\mathscr{L}(\text{vec}(W^t)) \\ &= \text{vec}(W^t) + \frac{\eta}{n}B^{t\top}(y - B^t\,\text{vec}(W^t+I)), \quad (3) \end{aligned}$$

where η is appropriately chosen step size and B^t is:

$$B^t = [v_1^*\mathbb{P}_1^tX \dots v_d^*\mathbb{P}_q^tX \dots v_d^*\mathbb{P}_d^tX],$$

where $\mathbb{P}_q^t = \text{diag}(\mathbb{1}_{X(w_1^t+e_1)\ge 0})$, for $q = \{1, \dots d\}$.

B. Alternating minimization

An alternating minimization-based approach can also be adopted to learn shallow networks with ReLU activations using a ℓ_2 loss function. At a high level, our algorithm rests on the following idea: given the knowledge of the correct signs of the input to each ReLU (B^*), the forward models can be *linearized*. Therefore, the weights $\text{vec}(W)$ can be estimated as in the case of any other linear model (e.g., via least-squares). The update rules are designed as follows:

$$B^{t'} = [v_1^*\text{diag}(\mathbb{1}_{X(w_1^{t'}+e_1)})X \dots v_d^*\text{diag}(\mathbb{1}_{X(w_d^{t'}+e_d)})X],$$

$$\mathrm{vec}(W^{t'+1}) = \arg\min_{\mathrm{vec}(W)} \left\| B^{t'} \cdot \mathrm{vec}(W+I) - y \right\|_2^2, \quad (4)$$

where $\mathbb{P}_q^{t'} = \mathrm{diag}(\mathbb{1}_{X(w_1^{t'}+e_1)\geq 0})$, for $q = \{1, \dots d\}$.

We now discuss our main theorem establishing convergence for both gradient descent (update rule (3)) and alternating minimization (update rule (4)).

C. Algorithmic guarantees

Our main theorem establishes that both gradient descent and alternating minimization can train a two layer ReLU network with linear convergence to the weights of the teacher network.

Theorem 1. *Given an initialization W^0 satisfying $\|W^0 - W^*\|_F \leq \delta \|W^* + \mathbf{I}\|_F$, for $0 < \delta < 1$, if we have number of training samples $n > C \cdot d \cdot k^2 \cdot poly(\log k, \log d, t)$, then with high probability $1 - ce^{-\alpha n} - d^{-\beta t}$, where c, α, β are positive constants and $t \geq 1$, the iterates of Gradient Descent* (3) *satisfy:*

$$\left\|W^{t+1} - W^*\right\|_F \leq \rho_{GD} \left\|W^t - W^*\right\|_F. \quad (5)$$

and the iterates of Alternating Minimization (4) *satisfy:*

$$\left\|W^{t+1} - W^*\right\|_F \leq \rho_{AM} \left\|W^t - W^*\right\|_F. \quad (6)$$

where and $0 < \rho_{AM} < \rho_{GD} < 1$.

Proof. We outline the proof technique for both methods as follows. We refer to Appendix A for key lemmas, which can be found in the full paper. Let $B^t := [v_1^* \mathbb{P}_1^t X, \dots, v_k^* \mathbb{P}_k^t X] := [A_1, \dots, A_k] = B$, where $\mathbb{P}_q^t = \mathrm{diag}(p_q^t)$, for $q = \{1, \dots k\}$. **Note:** Since we consider skipped connections, $k = d$.

a) Gradient descent: The update rule (3) requires:

$$\mathrm{vec}(W^{t+1}) = \mathrm{vec}(W^t) + \frac{\eta}{n} B^{t\top}(y - B^t \mathrm{vec}(W^t + I)).$$

Taking the difference between the learned weights and the weights of the teacher network,

$$\mathrm{vec}(W^{t+1}) - \mathrm{vec}(W^*) = \mathrm{vec}(W^{t+1}) - \mathrm{vec}(W^*) + \frac{\eta}{n} B^\top (y - B \,\mathrm{vec}(W^t + I)).$$

Taking the vector ℓ_2 norm on both sides,

$$\begin{aligned}
&\left\|W^{t+1} - W^*\right\|_{\mathrm{F}} \\
&\leq \left\|\mathbf{I} - \frac{\eta}{n}(B^\top B)\right\|_2 \left\|W^{t+1} - W^*\right\|_{\mathrm{F}} \\
&\qquad + \left\|\frac{B^\top}{\sqrt{n}}\right\|_2 \left\|\frac{1}{\sqrt{n}}(B^* - B)\,\mathrm{vec}(W^* + \mathbf{I})\right\|_2, \\
&\leq \frac{\sigma_{max}^2 - \sigma_{min}^2}{\sigma_{max}^2 + \sigma_{min}^2} \left\|W^{t+1} - W^*\right\|_{\mathrm{F}} + \eta\sigma_{max} \sum_{q=1}^k \|E_q\|_2, \\
&= \rho_4 \left\|W^t - W^*\right\|_{\mathrm{F}} + \eta\sigma_{max}\rho_3 \left\|W^t - W^*\right\|_{\mathrm{F}}, \\
&= \rho_{GD} \left\|W^t - W^*\right\|_{\mathrm{F}}, \qquad (7)
\end{aligned}$$

where $E_q := (B - B^*)\,\mathrm{vec}(W^* + \mathbf{I})/\sqrt{n}$ and $\sigma_{min}, \sigma_{max}$ are the minimum and maximum singular values of $\frac{B}{\sqrt{n}}$, respectively and are bounded via Lemma 1 in Appendix A, with probability $(1 - d^{-\beta t})$, as long as $n > C \cdot d \cdot k^2 \cdot poly(\log k, \log d, t)$. Also,

$$\left\|\mathbf{I} - \eta\frac{B^\top B}{n}\right\|_2 \leq \max\left(|1 - \eta\sigma_{min}^2|, |1 - \eta\sigma_{max}^2|\right) = \rho_4$$

and, we pick $\eta = 2/(\sigma_{min}^2 + \sigma_{max}^2)$ which minimizes parameter $\rho_{GD} = \rho_4 + \eta\sigma_{max}\rho_3$, for faster convergence.

Note that in utilizing this Lemma 1, we assume that the current weight matrix W^t, which composes matrix B, is independent of the samples X. This in general is not true, as the previous updates of W, also depends on the same data matrix. Hence this analysis only holds if fresh samples are drawn in each iteration of the algorithm.

Now, the error $\sum_q^k \|E_q\|_2$, is bounded as follows:

$$E_q = \frac{1}{\sqrt{n}}(A_q - A_q^*)(w_q^* + e_q) = \frac{v_q^*}{\sqrt{n}}(\mathbb{P}_q^t - \mathbb{P}_q^*)X(w_q^* + e_q).$$

$$\|E_q\|_2^2 = \sum_{i=1}^m (x_i^\top(w_q^* + e_q^*))^2 \cdot \frac{\mathbf{1}_{i,q}}{n} \leq \rho_{2,q}^2 \left\|w_q^t - w_q^*\right\|_2^2.$$

$$\sum_{q=1}^k \|E_q\|_2^2 \leq \sum_{q=1}^k \rho_{2,q}^2 \left\|w_q^t - w_q^*\right\|_2 \leq \max_q\left(\rho_{2,q}^2\right) \left\|W^t - W^*\right\|_{\mathrm{F}}^2,$$

where $\mathbf{1}_{i,q} := \mathbf{1}_{\left\{(x_i^\top(w_q^*+e_q^*))(x_i^\top(w_q^t+e_q))<0\right\}}$ is an indicator and we use $(v_q^*)^2 = 1$, the final bound is obtained via Corollary 1 in Appendix A, which holds with probability greater than $(1 - ce^{-\alpha n})$ as long as $n > C \cdot d \cdot k^2 \cdot \log k$. Subsequently,

$$\left(\sum_{q=1}^k \|E_q\|_2\right)^2 \leq k \sum_{q=1}^k \|E_q\|_2^2 \leq k \max_q\left(\rho_{2,q}^2\right) \left\|W^t - W^*\right\|_{\mathrm{F}}^2$$

$$\sum_{q=1}^k \|E_q\|_2 \leq \sqrt{k} \max_q\left(\rho_{2,q}\right) \left\|W^t - W^*\right\|_{\mathrm{F}} = \rho_3 \left\|W^t - W^*\right\|_{\mathrm{F}},$$

Each iteration of GD holds with probability $(1 - ce^{-\alpha n} - d^{-\beta t})$. Explicitly $\rho_{GD} = \frac{\kappa-1}{\kappa+1} + \frac{2\kappa\rho_3}{\sigma_{max}\cdot(\kappa+1)}$, with $\kappa = \frac{\sigma_{max}^2}{\sigma_{min}^2}$.

b) Alternating minimization: Since the minimization in (4) can be solved exactly, we get:

$$\begin{aligned}
\mathrm{vec}(W)^{t+1} &= (B^\top B)^{-1} B^\top y \\
&= (B^\top B)^{-1} B^\top B^* \,\mathrm{vec}(W^* + \mathbf{I}) \\
&= (B^\top B)^{-1} B^\top B \,\mathrm{vec}(W^* + \mathbf{I}) + \\
&\qquad (B^\top B)^{-1} B^\top (B^* - B)\,\mathrm{vec}(W^* + \mathbf{I}).
\end{aligned}$$

Taking the difference between the learned weights and the weights of the teacher network,

$$\mathrm{vec}(W)^{t+1} - \mathrm{vec}(W^*) = (B^\top B)^{-1} B^\top (B^* - B)\,\mathrm{vec}(W^* + \mathbf{I}).$$

Taking the vector ℓ_2 norm on both sides,

$$\begin{aligned}
&\left\|W^{t+1} - W^*\right\|_{\mathrm{F}} \\
&= \left\|(B^\top B)^{-1} B^\top (B^* - B^t)\,\mathrm{vec}(W^* + \mathbf{I})\right\|_2, \qquad (8) \\
&\leq \left\|n(B^\top B)^{-1}\right\|_2 \left\|\frac{B^\top}{\sqrt{n}}\right\|_2 \left\|\frac{1}{\sqrt{n}}(B^* - B^t)\,\mathrm{vec}(W^* + \mathbf{I})\right\|_2, \\
&\leq \frac{\sigma_{max}}{\sigma_{min}^2} \cdot \rho_3 \left\|W^t - W^*\right\|_{\mathrm{F}} < \rho_{AM} \left\|W^t - W^*\right\|_{\mathrm{F}} \qquad (9)
\end{aligned}$$

where, $\sigma_{min}, \sigma_{max}$ are the minimum and maximum singular values of $\frac{B}{\sqrt{n}}$, respectively and are bounded via Lemma 1 in Appendix A and $\sum_{q=1}^k \|E_q\|_2$ is bounded via Corollary 1 in Appendix A. Explicitly, $\rho_{GD} = \frac{\kappa\rho_3}{\sigma_{max}}$, with $\kappa = \frac{\sigma_{max}^2}{\sigma_{min}^2}$.

Comparison: $\rho_{GD} = \frac{\kappa-1}{\kappa+1} + \frac{2\rho_{AM}}{\kappa+1}$. Number of epochs T_{GD}

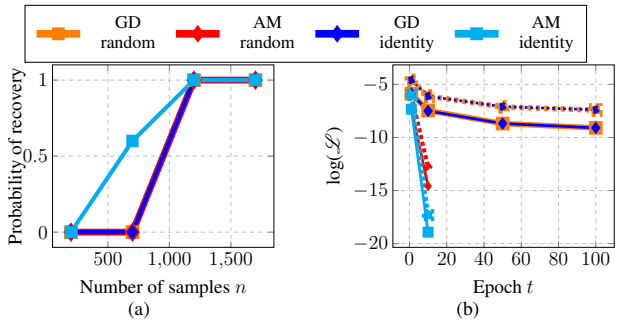

Fig. 1: (a) Successful parameter recovery for $d = 20$, with identity and random initializations; (b) training (solid) and testing (dotted) losses for fixed trial with $n = 1700$.

and T_{AM} required for ϵ-accuracy satisfy $\frac{T_{GD}}{T_{AM}} = \frac{\log(1/\rho_{AM})}{\log(1/\rho_{GD})}$. Therefore AM outperforms GD in terms of epoch complexity and enjoys faster convergence rate. □

D. Techniques for initialization

To prove convergence of both algorithms, we require that the initial weights W^0 are such that $\|W^0 - W^*\|_F \leq \delta \|W^* + \mathbf{I}\|_{\mathrm{F}}$ for small enough constant δ. A common assumption in the literature is that $\|W^*\|_{\mathrm{F}} \leq \gamma$ [11], [22], [24]. Now, suppose there exists some architecture satisfying $\|W^*\|_{\mathrm{F}} \leq \gamma \leq \frac{\delta\sqrt{d}}{1+\delta}$,

$$\delta \|W^* + \mathbf{I}\|_{\mathrm{F}} \geq \delta(\|\mathbf{I}\|_{\mathrm{F}} - \|W^*\|_{\mathrm{F}}) \geq \delta\sqrt{d} - \frac{\delta^2\sqrt{d}}{1+\delta} \geq \|W^*\|_{\mathrm{F}},$$

then $W^0 = \mathbf{0}$ satisfies the initialization requirement for Theorem 1. Additionally, we assume that the individual column norms of W^* are roughly balanced: $c(\frac{\gamma^2}{d}) \leq \|w_q^*\|_2^2 \leq C(\frac{\gamma^2}{d})$ for positive constants c, C for all q. □

IV. Experiments

For experimental validation, we select the training data $X \in \mathbb{R}^{n\times d}$, with $d = 20$, and entries picked from a normal $\mathcal{N}(0, 1/\sqrt{n})$ distribution. We use $y = f_{res}(W^*)$ to generate labels, with $\|W^*\|_{\mathrm{F}} = \gamma = 2$. We use identity initialization $W^0 = \mathbf{0}$, such that $\|W^*\|_{\mathrm{F}} \leq 0.8\|W^* + \mathbf{I}\|_{\mathrm{F}}$ for training the network with both AM and GD, and learn weights W^T at the end of T epochs. We compare the training against that with random initialization, where $\mathrm{vec}(W^0) = \mathcal{N}(0, \mathbf{I})/d$. The number of training samples n was swept from 200 to 1700 in steps of 500. We repeat this experiment over 10 trials with each trial being a random instantiation of X. Recovery is said to be successful if $\|W^T - W^*\|_F / \|W^* + \mathbf{I}\|_F < 0.01$ and the phase transition plot is presented in Fig. 1 (a), where the fraction of successful trials are plotted against number of training samples. We observe that the performance of our identity initialization matches that with random initialization, and GD and AM perform comparably. For $d = 20$, $n = 1200$ samples suffice for successful parameter recovery across all trials. AM also has lower epoch complexity (Fig. 1 (b)), requiring about 10 epochs to achieve the same training loss as that for GD in 1000 epochs. For $n = 1700$, we also plot test loss in 1 (b), where 500 test samples have entries from $\mathcal{N}(0, \mathbf{I}/\sqrt{500})$ and labels constructed according to $f_{res}(W^*)$. Constant step size η for GD is chosen by inspection.

References

[1] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in Neural Information Processing Systems*, pp. 586–594, 2016.

[2] M. Soltanolkotabi, A. Javanmard, and J. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Transactions on Information Theory*, 2018.

[3] K. Zhong, Z. Song, P. Jain, P. Bartlett, and I. Dhillon, "Recovery guarantees for one-hidden-layer neural networks," in *International Conference on Machine Learning*, pp. 4140–4149, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, IEEE, 2017.

[6] R. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[7] H. Lin and S. Jegelka, "Resnet with one-neuron hidden layers is a universal approximator," in *Advances in Neural Information Processing Systems*, pp. 6172–6181, 2018.

[8] H. Li, Z. Xu, G. Taylor, and T. Goldstein, "Visualizing the loss landscape of neural nets," *arXiv preprint arXiv:1712.09913*, 2017.

[9] I. Safran and O. Shamir, "Spurious local minima are common in two-layer relu neural networks," *arXiv preprint arXiv:1712.08968*, 2017.

[10] Y. Tian, "Symmetry-breaking convergence analysis of certain two-layered neural networks with relu nonlinearity," 2017.

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

[12] X. Zhang, Y. Yu, L. Wang, and Q. Gu, "Learning one-hidden-layer relu networks via gradient descent," *Proc. Int. Conf. Art. Intell. Stat. (AISTATS)*, 2018.

[13] S. Du, C. Jin, J. Lee, M. Jordan, A. Singh, and B. Poczos, "Gradient descent can take exponential time to escape saddle points," in *Advances in Neural Information Processing Systems*, pp. 1067–1077, 2017.

[14] O. Shamir, "Exponential convergence time of gradient descent for one-dimensional deep linear neural networks," *arXiv preprint arXiv:1809.08587*, 2018.

[15] S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

[16] P. Jain and P. Kar, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–336, 2017.

[17] H. Zhang and Y. Liang, "Reshaped wirtinger flow for solving quadratic system of equations," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, pp. 2622–2630, 2016.

[18] G. Jagatap and C. Hegde, "Fast, sample efficient algorithms for structured phase retrieval," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, pp. 4924–4934, 2017.

[19] G. Jagatap and C. Hegde, "Towards sample-optimal methods for solving random quadratic equations with structure," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2296–2300, IEEE, 2018.

[20] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Adv. Neural Inf. Proc. Sys. (NIPS)*, pp. 2796–2804, 2013.

[21] C. Ma, K. Wang, Y. Chi, and Y. Chen, "Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution," *arXiv preprint arXiv:1711.10467*, 2017.

[22] P. Bartlett, D. Helmbold, and P. Long, "Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks," *arXiv preprint arXiv:1802.06093*, 2018.

[23] A. Brutzkus and A. Globerson, "Globally optimal gradient descent for a convnet with gaussian inputs," in *International Conference on Machine Learning*, pp. 605–614, 2017.

[24] M. Hardt and T. Ma, "Identity matters in deep learning," *Int. Conf. Learning Representations (ICLR)*, 2017.