Width Provably Matters in Optimization for Deep Linear Neural Networks

Simon S. Du [*1] **Wei Hu** [*2]

Abstract

We prove that for an L-layer fully-connected linear neural network, if the width of every hidden layer is $\widetilde{\Omega}\left(L \cdot r \cdot d_{\text{out}} \cdot \kappa^3\right)$, where r and κ are the rank and the condition number of the input data, and d_{out} is the output dimension, then gradient descent with Gaussian random initialization converges to a global minimum at a linear rate. The number of iterations to find an ϵ-suboptimal solution is $O(\kappa \log(\frac{1}{\epsilon}))$. Our polynomial upper bound on the total running time for wide deep linear networks and the $\exp\left(\Omega\left(L\right)\right)$ lower bound for narrow deep linear neural networks [Shamir, 2018] together demonstrate that wide layers are necessary for optimizing deep models.

1. Introduction

Recent success in machine learning involves training deep neural networks using randomly initialized first order methods, which requires optimizing highly non-convex functions. Compared with nonlinear deep neural networks, deep linear networks are arguably more amenable to theoretical analysis. It is widely believed that deep linear networks already captures important aspects of optimization in deep learning (Saxe et al., 2014). Therefore, theoreticians have tried to study this problem in recent years. However, a strong global convergence guarantee is still missing.

A series of recent papers analyzed landscape of the deep linear network optimization problem (Kawaguchi, 2016; Hardt & Ma, 2016; Lu & Kawaguchi, 2017; Yun et al., 2017; Zhou & Liang, 2018; Laurent & Brecht, 2018). However, these results do not imply convergence of gradient-based methods to the global minimum. Recently, Bartlett et al. (2018); Arora et al. (2018a) directly analyzed the trajectory generated by gradient descent, and showed that gradient descent converges to global minimum under further assumptions on both data and global minimum. These results require specially designed initialization schemes, and do not apply to commonly used random initializations. In Section 2.1 we describe above results in more details.

A recent work by Shamir (2018) showed an exponential lower bound of randomly initialized gradient descent for narrow linear neural networks. More precisely, he showed that for an L-layer linear neural network in which the input, output and all hidden dimensions are equal to 1, gradient descent with Xavier initialization (Glorot & Bengio, 2010) requires at least $\exp\left(\Omega\left(L\right)\right)$ iteration to converge. This result demonstrates the intrinsic difficulty of optimizing deep networks: even in the basic setting, the convergence time for randomly initialized gradient descent can be exponential in depth. Nevertheless, this lower bound only holds for narrow neural networks. It is possible that making the hidden layers wider (which is usually the case in practice) can eliminate such exponential dependence on depth. This gives rise to the following questions:

*Can randomly initialized gradient descent optimize **wide** deep linear networks in polynomial time? If so, what is a sufficient width in hidden layers?*

Our Contribution: We answer the first question positively and give a concrete quantitative result for the second question. We prove that as long as the width of hidden layers is at least $\widetilde{\Omega}(L)$[1], gradient descent with Xavier initialization with high probability converges to the global minimum of the ℓ_2 loss at a linear rate *under no assumption*. To our knowledge, this is the first polynomial time global convergence guarantee for randomly initialized gradient descent for deep linear networks. Furthermore, our convergence rate is tight in the sense that it matches the convergence rate of applying gradient descent to the convex (1-layer) linear regression problem.

Compared with previous work (Bartlett et al., 2018; Arora et al., 2018a) that gave convergence rate guarantees for linear neural networks, our result has several advantages:

- Our result applies to the widely used Xavier random

[1]We omit dependence on other parameters here. See Theorem 4.1 for the precise requirements.

[*]Alphebetical order [1]Carnegie Mellon University, Pittsburgh, PA, USA [2]Princeton University, Princeton, NJ, USA. Correspondence to: Simon S. Du <ssdu@cs.cmu.edu>, Wei Hu <huwei@cs.princeton.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

initialization, while Bartlett et al. (2018) used identity initialization, and Arora et al. (2018a) assumed that initialization is "balanced" and somewhat close to the global minimum.

- Our result does not have any assumption on the input data, while Bartlett et al. (2018); Arora et al. (2018a) both required whitened data.
- Our result does not have any assumption on the global minimum, while Bartlett et al. (2018) assumed it to be either close to identity or positive definite, and Arora et al. (2018a) required it to have full rank.

Our polynomial upper bound for the wide linear neural network and the exponential lower bound for the narrow linear neural network together demonstrate that *width provably matters* in guaranteeing the efficiency of randomly initialized gradient descent for optimizing deep linear nets.[2]

Our Technique: Our proof technique is related to the recent work (Arora et al., 2018a;b; Du et al., 2018b) which utilized a time-varying Gram matrix (or preconditioner) along the trajectory of gradient descent. We adopt the same idea of using such Gram matrix. In the setting of wide linear neural networks, we carefully upper and lower bound eigenvalues of this Gram matrix throughout the optimization process, which together with some perturbation analysis implies linear convergence. In order to establish this at initialization, we need to analyze spectral properties of product of Gaussian random matrices and show that these properties hold throughout the trajectory of gradient descent.

2. Related Work

2.1. Optimization for Deep Linear Neural Networks

Landscape Analysis: Ge et al. (2015); Jin et al. (2017) showed that if an objective function satisfies that (1) all local minima are global, and (2) all saddle points are strict (i.e., there exists a negative curvature), then randomly perturbed gradient descent can escape all saddle points and find a global minimum. Motivated by this, a series of papers (Kawaguchi, 2016; Hardt & Ma, 2016; Lu & Kawaguchi, 2017; Yun et al., 2017; Zhou & Liang, 2018; Laurent & Brecht, 2018) studied these landscape properties for optimizing deep linear networks. While it was established that all local minima are global, unfortunately the strict saddle property is not satisfied even for 3-layer linear neural networks. Therefore, using landscape properties alone is not sufficient for proving global convergence.

[2]There are other techniques such as adaptive gradients (Duchi et al., 2011; Kingma & Ba, 2014) and skip-connections (He et al., 2016) that could help optimization. Analyses of those approaches are beyond the scope of this paper.

Trajectory Analysis: Instead of using the indirect landscape-based approach, an alternative is to directly analyze the trajectory generated by a concrete optimization algorithm like gradient descent. The current paper also belongs to this category.

Saxe et al. (2014) gave a thorough empirical study on deep linear networks, showing that they exhibit some learning patterns similar to nonlinear networks. Ji & Telgarsky (2019) studied the dynamics of gradient descent to optimize a deep linear neural network for classification problems, and showed that the risk converges to 0 and the solution found is a max-margin solution. Arora et al. (2018b) observed that adding more layers can accelerate optimization for certain loss functions. Du et al. (2018a) showed that using gradient descent, layers are automatically balanced.

All the above results do not show concrete convergence rates of gradient descent. The most related papers are (Bartlett et al., 2018) and (Arora et al., 2018a). Here we give a detailed description of their results.

Bartlett et al. (2018) showed that if one uses identity initialization, the input data is whitened, and the target matrix is either close to identity or positive definite, then gradient descent converges to the target matrix at a linear rate. Their result highly depends on the identity initialization scheme and has strong requirements on the input data and the target. Arora et al. (2018a) showed that if the initialization is balanced and the initial loss is smaller than the loss of any low-rank solution by a margin, then gradient descent converges to global minimum at a linear rate. However, their initialization scheme requires a special SVD step which is not used in practice, and the initial loss condition happens with exponentially small probability when the input and output dimensions are large. Our result improves upon these two papers by (i) allowing fully random initialization, and (ii) removing all assumptions on the input data and the target.

2.2. Optimization for Other Neural Networks

Many papers tried to identify the two desired geometric landscape properties of objective functions for non-linear neural networks (Freeman & Bruna, 2016; Nguyen & Hein, 2017; Venturi et al., 2018; Soudry & Carmon, 2016; Du & Lee, 2018; Soltanolkotabi et al., 2018; Haeffele & Vidal, 2017). Unfortunately, these properties do not hold even for simple non-linear shallow neural networks (Yun et al., 2019; Safran & Shamir, 2018).

A series of recent papers used trajectory-based methods to analyze gradient descent for shallow neural networks under strong data assumptions (Tian, 2017; Soltanolkotabi, 2017; Brutzkus & Globerson, 2017; Li & Yuan, 2017; Zhong et al., 2017; Zhang et al., 2018; Du et al., 2018c;d). These

results are restricted to shallow neural networks, and the assumptions are not satisfied in practice.

Recent breakthroughs were made in the optimization for extremely over-parametrized non-linear neural networks (Du et al., 2019; 2018b; Li & Liang, 2018; Allen-Zhu et al., 2018; Zou et al., 2018). For deep ReLU neural networks, Allen-Zhu et al. (2018); Zou et al. (2018) showed that if the width of hidden layers is $\Omega\left(n^{30}L^{30}\log^2(\frac{1}{\epsilon})\right)$, then gradient descent converges to ϵ loss. (n is the number of training samples.) Du et al. (2018b) considered non-linear smooth activation functions like soft-plus, and showed that if the width of hidden layers is $n^4 \cdot 2^{\Omega(L)}$, then gradient descent converges to 0 loss.[3] All these results need additional assumptions on data, which also show up in the required width. Compared with them, we have a much better bound on the required width (L v.s. L^{30} or $\exp(L)$), although this is not a fair comparison because linear networks are simpler than non-linear ones. But given that we obtain a near linear dependence on depth, our result may shed light on the limit of required width in optimizing non-linear neural networks.

3. Preliminaries

3.1. Notation

We use $\|\cdot\|$ to denote the Euclidean norm of a vector or the spectral norm of a matrix, and use $\|\cdot\|_F$ to denote the Frobenius norm of a matrix. For a symmetric matrix, let $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ be its maximum and minimum eigenvalues, and let $\lambda_i(A)$ be its i-th largest eigenvalue. Similarly, for a general matrix B, let $\sigma_{\max}(B)$ and $\sigma_{\min}(B)$ be its maximum and minimum singular values, and let $\sigma_i(B)$ be its i-th largest singular value.

Let I be the identity matrix and $[n] = \{1, 2, \ldots, n\}$. Denote by $\mathcal{N}(0,1)$ the standard Gaussian distribution, and by χ^2_k the χ^2 distribution with k degrees of freedom. Let $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$ be the unit sphere in $\mathbb{R}^d$.

Let $\mathrm{vec}(A)$ be the vectorization of a matrix A in column-first order. The Kronecker product between two matrices $A \in \mathbb{R}^{m_A \times n_A}$ and $B \in \mathbb{R}^{m_B \times n_B}$ is defined as

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \cdots & a_{1,n_A}B \\ \vdots & \ddots & \vdots \\ a_{m_A,1}B & \cdots & a_{m_A,n_A}B \end{pmatrix} \in \mathbb{R}^{m_A m_B \times n_A n_B},$$

where $a_{i,j}$ is the element in the (i,j)-th entry of A.

We use C to represent a sufficiently large universal constant throughout the paper. The specific value of C can be different from line to line.

[3] They also showed if one uses skip-connections (He et al., 2016), then the width only depends polynomially on L. We only focus on fully-connected neural networks in this paper.

3.2. Problem Setup

We are given n training samples $\{(x_p, y_p)\}_{p=1}^n \subset \mathbb{R}^{d_{\text{in}}} \times \mathbb{R}^{d_{\text{out}}}$. Let $X = (x_1, \ldots, x_n) \in \mathbb{R}^{d_{\text{in}} \times n}$ be the input data matrix and $Y = (y_1, \ldots, y_n) \in \mathbb{R}^{d_{\text{out}} \times n}$ be the label matrix.

Consider the problem of training a depth-L linear neural network with hidden layer width m by minimizing the ℓ_2 loss over data:

$$\begin{aligned}\ell(W_1, \ldots, W_L) &= \frac{1}{2}\sum_{p=1}^n \left\| \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} W_L \cdots W_1 x_p - y_p \right\|^2 \\ &= \frac{1}{2}\left\| \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} W_L \cdots W_1 X - Y \right\|_F^2, \end{aligned} \tag{1}$$

where $W_1 \in \mathbb{R}^{m \times d_{\text{in}}}, W_2, \ldots, W_{L-1} \in \mathbb{R}^{m \times m}$ and $W_L \in \mathbb{R}^{m \times d_{\text{out}}}$ are weight matrices to be learned. Here $\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}$ is a scaling factor corresponding to Xavier initialization[4] (Glorot & Bengio, 2010), for which we provide a justification in Section 3.3.

We consider the vanilla gradient descent (GD) algorithm for objective (1) with random initialization:

- We initialize all the entries of $W_1, \ldots, W_L$ independently from $\mathcal{N}(0,1)$. Let $W_1(0), \ldots, W_L(0)$ be the weight matrices at initialization.
- Then we update the weights using GD: for $t = 0, 1, 2, \ldots$ and $i \in [L]$,

$$W_i(t+1) = W_i(t) - \eta \frac{\partial \ell}{\partial W_i}(W_1(t), \ldots, W_L(t)), \tag{2}$$

where $\eta > 0$ is the learning rate.

For notational convenience, we denote $W_{j:i} = W_j W_{j-1} \cdots W_i$ for every $1 \le i \le j \le L$. We also define $W_{i-1:i} = I$ (of appropriate dimension) for completeness.

We use the time index t for all variables that depend on $W_1, \ldots, W_L$, e.g., $W_{j:i}(t) = W_j(t) \cdots W_i(t)$, $\ell(t) = \ell(W_1(t), \ldots, W_L(t))$, etc.

3.3. On the Scaling Factor

The scaling factor $\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}$ ensures that the network at initialization preserves the size of every input in expectation.

Claim 3.1. *For any $x \in \mathbb{R}^{d_{\text{in}}}$, we have*

$$\mathbb{E}\left[\left\| \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} W_{L:1}(0) x \right\|^2\right] = \|x\|^2.$$

The proof of Claim 3.1 is given in Appendix A.

[4] We adopt this scaling factor so that we can initialize all weights from $\mathcal{N}(0,1)$.

4. Main Result

In this section we present our main result. First note that when $m \geq d_{\text{out}}$ (which we will assume), the deep linear network we study has the same representation power as a linear map $x \mapsto Wx$ ($W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$). Hence, the optimal value OPT for our objective function (1) is equal to the optimal value of the following linear regression problem:

$$\mathsf{OPT} = \min_{W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}} f(W) = \min_{W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}} \frac{1}{2} \|WX - Y\|_F^2 . \tag{3}$$

Let $\Phi \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ be a minimizer of f with minimum spectral norm.[5] Let $r = \text{rank}(X)$, and define $\kappa = \frac{\lambda_{\max}(X^\top X)}{\lambda_r(X^\top X)}$ which is the condition number of $X^\top X$.

Our main theorem is the following:

Theorem 4.1. *Suppose*

$$m \geq C \cdot L \cdot \max\left\{ r\kappa^3 d_{\text{out}}(1 + \|\Phi\|^2), r\kappa^3 \log\frac{r}{\delta}, \log L \right\} \tag{4}$$

for some $\delta \in (0,1)$ and a sufficiently large universal constant $C > 0$ and we set $\eta \leq \frac{d_{\text{out}}}{3L\|X^\top X\|}$. Then with probability at least $1 - \delta$ over the random initialization, we have

$$\ell(0) - \mathsf{OPT} \leq O\left(\max\left\{ 1, \frac{\log(r/\delta)}{d_{\text{out}}}, \|\Phi\|^2 \right\} \right) \|X\|_F^2 ,$$

$$\ell(t) - \mathsf{OPT} \leq \left(1 - \frac{\eta L \cdot \lambda_r(X^\top X)}{4 d_{\text{out}}} \right)^t (\ell(0) - \mathsf{OPT}) .$$

Theorem 4.1 establishes that if the width of each layer is sufficiently large, randomly initialized gradient descent can reach a global minimum at a linear convergence rate. Notably, our result is fully polynomial in the sense that we only require polynomially large width and the convergence time is also polynomial. To our knowledge, this is the first polynomial time convergence guarantee for randomly initialized gradient descent on deep linear networks.

Ignoring logarithmic factors and assuming $\|\Phi\| = O(1)$, our requirement on width (4) is $m = \widetilde{\Omega}\left(Lr\kappa^3 d_{\text{out}}\right)$. It remains open whether this dependence is tight for randomly initialized gradient descent to find a global minimum in polynomial time.

In terms of convergence rate, if we set the learning rate to be $\eta = \Theta\left(\frac{d_{\text{out}}}{L\|X^\top X\|}\right)$, then the predicted ratio of decrease in each iteration is $1 - \Theta\left(\frac{\lambda_r(X^\top X)}{\|X^\top X\|}\right) = 1 - \Theta\left(\frac{1}{\kappa}\right)$, so the number of iterations needed to reach $\mathsf{OPT} + \epsilon$ loss is $O\left(\kappa \log \frac{1}{\epsilon}\right)$. This matches the convergence rate of gradient descent on the linear regression (convex!) problem (3).

[5] Our theorem holds for any minimizer of f. Since our bound improves when $\|\Phi\|$ is smaller, we simply define Φ to be a minimum-spectral-norm minimizer.

Furthermore, notice that our requirement on the learning rate is $\eta = O(\frac{d_{\text{out}}}{L\|X^\top X\|})$. When $L = 1$, this also exactly recovers the convergence result for applying gradient descent to the linear regression problem (3). The reason why L is in the denominator will be clear in the proof. At a high level, we show that optimizing a deep linear network is similar to a linear regression problem with the covariance matrix being $L \cdot X^\top X$, which thus requires scaling down the learning rate by a factor of L.

5. Proof Overview

In this section we give an overview for the proof of Theorem 4.1.

First, we note that a simple reduction implies that we can make the following assumption *without loss of generality*:

Assumption 5.1. *(Without loss of generality) $X \in \mathbb{R}^{d_{\text{in}} \times r}$, $\text{rank}(X) = r$, $Y = \Phi X$, and $\mathsf{OPT} = 0$.*

See Appendix B for justification. Therefore we will work under Assumption 5.1 from now on.

Now we proceed to sketch the proof of Theorem 4.1. The key idea is to examine the dynamics of the network prediction on data X during optimization, namely:

$$U = \frac{1}{\sqrt{m^{L-1} d_{\text{out}}}} W_{L:1} X \in \mathbb{R}^{d_{\text{out}} \times n}.$$

With this notation, the network prediction at iteration t is $U(t) = \frac{1}{\sqrt{m^{L-1} d_{\text{out}}}} W_{L:1}(t) X$, and the loss value at iteration t is $\ell(t) = \frac{1}{2}\|U(t) - Y\|_F^2$. Hence how $U(t)$ evolves is directly related to how loss $\ell(t)$ decreases.

The gradient of our objective function (1) is

$$\frac{\partial \ell}{\partial W_i} = \frac{1}{\sqrt{m^{L-1} d_{\text{out}}}} W_{L:i+1}^\top (U - Y) \left(W_{i-1:1} X\right)^\top . \tag{5}$$

Then using the update rule (2) we write

$$\begin{aligned}
&W_{L:1}(t+1) \\
=& \prod_i \left(W_i(t) - \eta \frac{\partial \ell}{\partial W_i}(t) \right) \\
=& W_{L:1}(t) - \sum_{i=1}^{L} \eta W_{L:i+1}(t) \frac{\partial \ell}{\partial W_i}(t) W_{i-1:1}(t) + E(t) \\
=& W_{L:1}(t) \\
&- \frac{\eta}{\sqrt{m^{L-1} d_{\text{out}}}} \sum_{i=1}^{L} \Big(W_{L:i+1}(t) W_{L:i+1}^\top(t) \\
&\cdot (U(t) - Y) \left(W_{i-1:1}(t) X\right)^\top W_{i-1:1}(t) \Big) \\
&+ E(t),
\end{aligned}$$

where $E(t)$ contains all high-order terms (those with η^2 or higher). Multiplying this equation by $\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}X$ on the right we get

$$U(t+1) = U(t) - \frac{\eta}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \Big(W_{L:i+1}(t) W_{L:i+1}^\top(t) \cdot (U(t) - Y) \left(W_{i-1:1}(t)X\right)^\top \left(W_{i-1:1}(t)X\right) \Big) + \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} E(t)X.$$

Vectorizing the above equation and using the property of Kronecker product: $\text{vec}\,(ACB) = (B^\top \otimes A)\text{vec}\,(C)$, we obtain

$$\begin{aligned} &\text{vec}\,(U(t+1) - U(t)) \\ &= -\eta P(t) \cdot \text{vec}\,(U(t) - Y) + \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} \text{vec}\,(E(t)X), \end{aligned} \tag{6}$$

where

$$P(t) = \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \Big[\Big(\left(W_{i-1:1}(t)X\right)^\top \left(W_{i-1:1}(t)X\right) \Big) \otimes \left(W_{L:i+1}(t) W_{L:i+1}^\top(t)\right) \Big]. \tag{7}$$

Notice that $P(t)$ is always positive semi-definite (PSD) because it is the sum of L terms, each of which is the Kronecker product between two PSD matrices.

Now we assume that the high-order term $E(t)$ in (6) is very small (which we will rigorously prove) and ignore it for now. Then (6) implies

$$\text{vec}\,(U(t+1) - Y) \approx (I - \eta P(t))\,\text{vec}\,(U(t) - Y). \tag{8}$$

Suppose we are able to set $\eta \le \frac{1}{\lambda_{\max}(P_t)}$. Then (8) would imply

$$\|U(t+1) - Y\|_F \le (1 - \eta\lambda_{\min}(P(t)))\,\|U(t) - Y\|_F.$$

Therefore, if we have a lower bound on $\lambda_{\min}(P(t))$ for all t, we will have linear convergence as desired. We will indeed prove the following bounds on $\lambda_{\max}(P_t)$ and $\lambda_{\min}(P_t)$ for all t, which will essentially complete the proof:

$$\begin{aligned} \lambda_{\max}(P_t) &\le O\left(L\lambda_{\max}(X^\top X)/d_{\text{out}}\right), \\ \lambda_{\min}(P_t) &\ge \Omega\left(L\lambda_{\min}(X^\top X)/d_{\text{out}}\right). \end{aligned} \tag{9}$$

We use the following approach to bound $\lambda_{\max}(P(t))$ and $\lambda_{\min}(P(t))$:

$$\begin{aligned} &\lambda_{\max}(P(t)) \\ &\le \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \Big[\lambda_{\max}\left(\left(W_{i-1:1}(t)X\right)^\top \left(W_{i-1:1}(t)X\right) \right) \cdot \lambda_{\max}\left(W_{L:i+1}(t) W_{L:i+1}^\top(t) \right) \Big] \\ &= \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \sigma_{\max}^2\left(W_{i-1:1}(t)X\right) \cdot \sigma_{\max}^2\left(W_{L:i+1}(t)\right), \\ &\lambda_{\min}(P(t)) \\ &\ge \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \Big[\lambda_{\min}\left(\left(W_{i-1:1}(t)X\right)^\top \left(W_{i-1:1}(t)X\right) \right) \cdot \lambda_{\min}\left(W_{L:i+1}(t) W_{L:i+1}^\top(t) \right) \Big] \\ &= \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \sigma_{\min}^2\left(W_{i-1:1}(t)X\right) \cdot \sigma_{\min}^2\left(W_{L:i+1}(t)\right), \end{aligned} \tag{10}$$

Here we have used the property that for symmetric matrices A and B, every eigenvalue of $A \otimes B$ is the product of an eigenvalue of A and an eigenvalue of B. Therefore, it suffices to obtain upper and lower bounds on the singular values of $W_{i-1:1}(t)X$ and $W_{L:i+1}(t)$. In Section 6, we establish these bounds for initialization ($t = 0$). Then we finish the proof of Theorem 4.1 in Section 7.

6. Properties at Initialization

In this section we establish some properties of the weight matrices generated by random initialization.

The following lemma shows that when multiplying a fixed vector by a series of Gaussian matrices with large width, the resulting vector's norm is concentrated.

Lemma 6.1. *Suppose $m > Cq$, and consider q independent random matrices $A_1, \ldots, A_q$ ($A_1 \in \mathbb{R}^{m\times d}, A_2, \ldots, A_q \in \mathbb{R}^{m\times m}$) with i.i.d. $\mathcal{N}(0,1)$ entries. Then for any $v \in \mathcal{S}^{d-1}$, with probability at least $1 - e^{-\Omega(m/q)}$ we have*

$$0.9m^{q/2} \le \|A_q \cdots A_1 v\| \le 1.1m^{q/2}.$$

Proof. See Appendix C. □

The next three propositions show the key properties of products of weight matrices at initialization.

Proposition 6.2. *For any $1 < i \le L$, with probability at least $1 - e^{-\Omega(m/L)}$ we have*

$$\begin{aligned} \sigma_{\max}(W_{L:i}(0)) &\le 1.2m^{\frac{L-i+1}{2}}, \\ \sigma_{\min}\left(W_{L:i}(0)\right) &\ge 0.8m^{\frac{L-i+1}{2}}. \end{aligned}$$

Proof. Let $A = W_{L:i}^\top(0)$. Since $A \in \mathbb{R}^{m\times d_{\text{out}}}$ and $m > d_{\text{out}}$, we know $\|A\| = \sup_{v\in\mathcal{S}^{d_{\text{out}}-1}} \|Av\|$ and $\sigma_{\min}(A) =$

$\inf_{v\in\mathcal{S}^{d_{\text{out}}-1}}\|Av\|$. Also, from Lemma 6.1 we know that for any fixed $v\in\mathcal{S}^{d_{\text{out}}-1}$, with probability at least $1-e^{-\Omega(m/L)}$ we have $\|Av\|\in\left[0.9m^{\frac{L-i+1}{2}},1.1m^{\frac{L-i+1}{2}}\right]$.

The rest of the proof is by a standard ε-net argument. Let $\varepsilon=0.01$. Take an ε-net $\mathcal{N}$ for $\mathcal{S}^{d_{\text{out}}-1}$ with $|\mathcal{N}|\le(3/\varepsilon)^{d_{\text{out}}}$. By a union bound, with probability at least $1-|\mathcal{N}|\,e^{-\Omega(m/L)}$, for all $u\in\mathcal{N}$ simultaneously we have $\|Au\|/\|u\|\in\left[0.9m^{\frac{L-i+1}{2}},1.1m^{\frac{L-i+1}{2}}\right]$. Suppose this happens for every $u\in\mathcal{N}$. Next, for any $v\in\mathcal{S}^{d_{\text{out}}-1}$, there exists $u\in\mathcal{N}$ such that $\|u-v\|\le\varepsilon$. Then we have

$$\begin{aligned}\|Av\|&\le\|Au\|+\|A(u-v)\|\le1.1m^{\frac{L-i+1}{2}}\|u\|+\varepsilon\|A\|\\&\le1.1(1+\varepsilon)m^{\frac{L-i+1}{2}}+\varepsilon\|A\|.\end{aligned}$$

Taking supreme over $v\in\mathcal{S}^{d_{\text{out}}-1}$, we obtain

$$\|A\|\le\frac{1.1(1+\varepsilon)m^{\frac{L-i+1}{2}}}{1-\varepsilon}\le1.2m^{\frac{L-i+1}{2}}.$$

For the lower bound, we have

$$\begin{aligned}\|Av\|&\ge\|Au\|-\|A(u-v)\|\ge0.9m^{\frac{L-i+1}{2}}\|u\|-\varepsilon\|A\|\\&\ge0.9(1-\varepsilon)m^{\frac{L-i+1}{2}}-\varepsilon\cdot1.2m^{\frac{L-i+1}{2}}\\&\ge0.8m^{\frac{L-i+1}{2}}.\end{aligned}$$

Taking infimum over $v\in\mathcal{S}^{d_{\text{out}}-1}$ we get $\sigma_{\min}(A)\ge0.8m^{\frac{L-i+1}{2}}$.

The success probability is at least $1-|\mathcal{N}|\,e^{-\Omega(m/L)}=1-e^{-\Omega(\frac{m}{L})+d_{\text{out}}\log(\frac{3}{\varepsilon})}=1-e^{-\Omega(\frac{m}{L})}$ since $m>CLd_{\text{out}}$. □

Proposition 6.3. *For any $1\le i<L$, with probability at least $1-e^{-\Omega(m/L)}$ we have*

$$\begin{aligned}\sigma_{\max}(W_{i:1}(0)\cdot X)&\le1.2m^{\frac{i}{2}}\sigma_{\max}(X),\\\sigma_{\min}(W_{i:1}(0)\cdot X)&\ge0.8m^{\frac{i}{2}}\sigma_{\min}(X).\end{aligned}$$

Proof. The proof is similar to the proof of Proposition 6.2 and is deferred to Appendix D. □

Proposition 6.4. *For any $1<i\le j<L$, with probability at least $1-e^{-\Omega(m/L)}$ we have*

$$\|W_{j:i}(0)\|\le O\left(\sqrt{L}m^{\frac{j-i+1}{2}}\right).$$

Proof. Let $A=W_{j:i}(0)$. From Lemma 6.1 we know that for any fixed $v\in\mathcal{S}^{m-1}$, with probability at least $1-e^{-\Omega(m/L)}$ we have $\|Av\|\in\left[0.9m^{\frac{j-i+1}{2}},1.1m^{\frac{j-i+1}{2}}\right]$.

Take a small constant $c>0$ and partition the index set $[m]$ into $[m]=S_1\cup S_2\cup\cdots\cup S_{L/c}$ where $|S_l|=cm/L$ for each l. For each l, taking a $\frac{1}{2}$-net $\mathcal{N}_l$ for all the unit vectors supported in S_l, i.e., a $\frac{1}{2}$-net for the set $V_{S_l}=\left\{v\in\mathcal{S}^{m-1}:\text{supp}(v)\subseteq S_l\right\}$, we know that

$$\|Au\|\le O\left(m^{\frac{j-i+1}{2}}\right),\quad\forall u\in V_{S_l},\tag{11}$$

with probability at least $1-|\mathcal{N}_l|\,e^{-\Omega(m/L)}=1-e^{-\Omega(m/L)+(cm/L)\log6}=1-e^{-\Omega(m/L)}$. Then taking a union bound over all l, we know that (11) holds for all l simultaneously with probability at least $1-\frac{L}{c}e^{-\Omega(m/L)}$. Conditioned on this, for any $v\in\mathbb{R}^m$, we can partition its coordinates and write it as the sum $v=\alpha_1v_1+\alpha_2v_2+\cdots+\alpha_{L/c}v_{L/c}$ where $\alpha_l\in\mathbb{R}$ and $v_l\in V_{S_l}$ for each l. Then we have

$$\begin{aligned}\|Av\|&\le\sum\nolimits_l\|A\cdot\alpha_lv_l\|\le\sum\nolimits_l|\alpha_l|\,O\left(m^{\frac{j-i+1}{2}}\right)\\&\le O\left(m^{\frac{j-i+1}{2}}\right)\sqrt{\frac{L}{c}\sum\nolimits_l\alpha_l^2}=O\left(\sqrt{L}m^{\frac{j-i+1}{2}}\right)\|v\|.\end{aligned}$$

This means $\|A\|\le O\left(\sqrt{L}m^{\frac{j-i+1}{2}}\right)$. The success probability is at least $1-\frac{L}{c}e^{-\Omega(m/L)}=1-e^{-\Omega(m/L)+\log(L/c)}=1-e^{-\Omega(m/L)}$ since $m>CL\log L$. □

To close this section, we bound the loss value $\ell(0)$ at initialization, which proves the first part of Theorem 4.1.

Proposition 6.5. *With probability at least $1-e^{-\Omega(m/L)}-\delta/2$, we have $\ell(0)\le O\left(\max\left\{1,\frac{\log(r/\delta)}{d_{\text{out}}},\|\Phi\|^2\right\}\right)\|X\|_F^2$.*

Proof. See Appendix E. □

7. Proof of the Main Theorem

In this section we prove Theorem 4.1 based on ingredients from Sections 5 and 6.

From Propositions 6.2, 6.3, 6.4 and 6.5, we know that with probability at least $1-L^2e^{-\Omega(m/L)}-\delta/2\ge1-\delta$, the following conditions of initialization are satisfied simultaneously:

$$\begin{cases}\sigma_{\max}(W_{L:i}(0))\le1.2m^{\frac{L-i+1}{2}},&\forall1<i\le L,\\\sigma_{\min}(W_{L:i}(0))\ge0.8m^{\frac{L-i+1}{2}},&\forall1<i\le L,\\\sigma_{\max}(W_{i:1}(0)\cdot X)\le1.2m^{\frac{i}{2}}\sigma_{\max}(X),&\forall1\le i<L,\\\sigma_{\min}(W_{i:1}(0)\cdot X)\ge0.8m^{\frac{i}{2}}\sigma_{\min}(X),&\forall1\le i<L,\\\|W_{j:i}(0)\|\le O\left(\sqrt{L}m^{\frac{j-i+1}{2}}\right),&\forall1<i\le j<L,\\\ell(0)\le B.\end{cases}\tag{12}$$

Here we define $B=O\left(\max\left\{1,\frac{\log(r/\delta)}{d_{\text{out}}},\|\Phi\|^2\right\}\right)\|X\|_F^2$ which is the upper bound on $\ell(0)$ from Proposition 6.5.

From our requirement on m (4), we know

$$m \geq C \cdot Lr\kappa^3 \max\left\{d_{\text{out}}(1+\|\Phi\|^2), \log\frac{r}{\delta}\right\} \geq \frac{CLr\kappa^3 d_{\text{out}}B}{\|X\|_F^2} \geq \frac{CLr\kappa^3 d_{\text{out}}B}{r\|X\|^2} = \frac{CL\|X\|^4 d_{\text{out}}B}{\sigma_{\min}^6(X)}. \tag{13}$$

Now we establish our convergence result conditioned on all properties in (12). Specifically, we use induction on t to simultaneously prove the following three properties $\mathcal{A}(t)$, $\mathcal{B}(t)$ and $\mathcal{C}(t)$ for all $t = 0, 1, \ldots$:

- $\mathcal{A}(t)$:
$$\ell(t) \leq \left(1 - \frac{1}{4}\eta L\sigma_{\min}^2(X)/d_{\text{out}}\right)^t \ell(0) \leq \left(1 - \frac{1}{4}\eta L\sigma_{\min}^2(X)/d_{\text{out}}\right)^t B.$$

- $\mathcal{B}(t)$:
$$\begin{cases} \sigma_{\max}(W_{L:i}(t)) \leq \frac{5}{4}m^{\frac{L-i+1}{2}}, \forall 1 < i \leq L, \\ \sigma_{\min}(W_{L:i}(t)) \geq \frac{3}{4}m^{\frac{L-i+1}{2}}, \forall 1 < i \leq L, \\ \sigma_{\max}(W_{i:1}(t)\cdot X) \leq \frac{5}{4}m^{\frac{i}{2}}\sigma_{\max}(X), \forall 1 \leq i < L, \\ \sigma_{\min}(W_{i:1}(t)\cdot X) \geq \frac{3}{4}m^{\frac{i}{2}}\sigma_{\min}(X), \forall 1 \leq i < L, \\ \|W_{j:i}(t)\| \leq O\left(\sqrt{L}m^{\frac{j-i+1}{2}}\right), \forall 1 < i \leq j < L. \end{cases}$$

- $\mathcal{C}(t)$:
$$\|W_i(t) - W_i(0)\|_F \leq \frac{24\sqrt{Bd_{\text{out}}}\|X\|}{L\sigma_{\min}^2(X)}, \quad \forall i \in [L].$$

Notice that if we prove $\mathcal{A}(t)$ for all $t \geq 0$, we will finish the proof of Theorem 4.1.

The initial conditions $\mathcal{A}(0)$ and $\mathcal{B}(0)$ follow directly from (12), and $\mathcal{C}(0)$ is trivially true. In order to establish $\mathcal{A}(t)$, $\mathcal{B}(t)$ and $\mathcal{C}(t)$ for all t, in Sections 7.1-7.3 we will prove respectively the following claims for all $t \geq 0$:

Claim 7.1. $\mathcal{A}(0), \ldots, \mathcal{A}(t), \mathcal{B}(0), \ldots, \mathcal{B}(t) \Longrightarrow \mathcal{C}(t+1)$.

Claim 7.2. $\mathcal{C}(t) \Longrightarrow \mathcal{B}(t)$.

Claim 7.3. $\mathcal{A}(t), \mathcal{B}(t) \Longrightarrow \mathcal{A}(t+1)$.

The proof of Theorem 4.1 is finished after the above three claims are proved.

7.1. Proof of Claim 7.1

Denote $\gamma = \frac{1}{4}L\sigma_{\min}^2(X)/d_{\text{out}}$. From $\mathcal{A}(0), \ldots, \mathcal{A}(t)$ we know $\ell(s) \leq (1-\eta\gamma)^s B$ for all $0 \leq s \leq t$.

From the gradient expression (5), for all $0 \leq s \leq t$ and all $i \in [L]$ we can bound:

$$\begin{aligned} &\left\|\frac{\partial \ell}{\partial W_i}(s)\right\|_F \\ &\leq \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}\|W_{L:i+1}(s)\|\,\|U(s)-Y\|_F\,\|W_{i-1:1}(s)X\| \\ &\leq \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}\cdot\frac{5}{4}m^{\frac{L-i}{2}}\cdot\sqrt{2\ell(s)}\cdot\frac{5}{4}m^{\frac{i-1}{2}}\|X\| \\ &\leq \frac{3\sqrt{(1-\eta\gamma)^s B}}{\sqrt{d_{\text{out}}}}\|X\|, \end{aligned} \tag{14}$$

where we have used $\mathcal{B}(s)$.

Then we can bound $\|W_i(t+1) - W_i(0)\|_F$ for all $i \in [n]$:

$$\begin{aligned} &\|W_i(t+1) - W_i(0)\|_F \leq \sum_{s=0}^{t}\|W_i(s+1) - W_i(s)\|_F \\ &= \sum_{s=0}^{t}\left\|\eta\frac{\partial \ell}{\partial W_i}(s)\right\|_F \leq \eta\sum_{s=0}^{t}\frac{3\sqrt{(1-\eta\gamma)^s B}}{\sqrt{d_{\text{out}}}}\|X\| \\ &\leq \frac{3\eta\sqrt{B}}{\sqrt{d_{\text{out}}}}\|X\|\sum_{s=0}^{t-1}(1-\eta\gamma/2)^s \leq \frac{3\eta\sqrt{B}}{\sqrt{d_{\text{out}}}}\|X\|\cdot\frac{2}{\eta\gamma} \\ &= \frac{24\sqrt{Bd_{\text{out}}}\|X\|}{L\sigma_{\min}^2(X)}. \end{aligned}$$

This proves $\mathcal{C}(t+1)$.

7.2. Proof of Claim 7.2

Let $R = \frac{24\sqrt{Bd_{\text{out}}}\|X\|}{L\sigma_{\min}^2(X)}$ and denote $\Delta_i = W_i(t) - W_i(0)$ $(i \in [L])$. Then using $\|\Delta_i\|_F \leq R\,(\forall i \in [L])$ we will show the followings:

$$\|W_{L:i}(t) - W_{L:i}(0)\| \leq 0.05m^{\frac{L-i+1}{2}}, \forall 1 < i \leq L, \tag{15}$$

$$\|(W_{i:1}(t) - W_{i:1}(0))X\| \leq 0.05m^{\frac{i}{2}}\sigma_{\min}(X), \forall 1 \leq i < L, \tag{16}$$

$$\|W_{j:i}(t) - W_{j:i}(0)\| \leq 0.05\sqrt{L}m^{\frac{j-i+1}{2}}, \forall 1 < i \leq j < L. \tag{17}$$

Combing them with (12), we will finish the proof of $\mathcal{B}(t)$.

First we prove (17). For $1 \leq i < j \leq L$, we can write

$$W_{j:i}(t) = (W_j(0) + \Delta_j)\cdots(W_i(0) + \Delta_i).$$

Expanding the above product, each term except the leading term $W_{j:i}(0)$ has the form:

$$W_{j:(k_s+1)}(0)\cdot\Delta_{k_s}\cdot W_{(k_s-1):(k_{s-1}+1)}(0)\cdot\Delta_{k_{s-1}}\cdots\cdot\Delta_{k_1}\cdot W_{(k_1-1):i}(0), \tag{18}$$

where $i \le k_1 < \cdots < k_s \le j$ are positions at which perturbation terms Δ_{k_l} are taken out, and at any other position k, $W_k(0)$ is used. Note that every factor in (18) of the form $W_{b:a}(0)$ satisfies $\|W_{b:a}(0)\| \le O\left(\sqrt{L}m^{\frac{b-a+1}{2}}\right)$ because of (12); there are at most $s+1$ such factors, so the product of their spectral norms is at most $\left(O(\sqrt{L})\right)^{s+1} m^{\frac{j-i+1-s}{2}}$. Thus, we can bound the sum of all terms like (18) as

$$\begin{aligned}
&\|W_{j:i}(t) - W_{j:i}(0)\| \\
\le& \sum_{s=1}^{j-i+1} \binom{j-i+1}{s} R^s \left(O\left(\sqrt{L}\right)\right)^{s+1} m^{\frac{j-i+1-s}{2}} \\
\le& \sum_{s=1}^{j-i+1} L^s R^s \left(O\left(\sqrt{L}\right)\right)^{s+1} m^{\frac{j-i+1-s}{2}} \\
=& O\left(\sqrt{L}\right) \sum_{s=1}^{j-i+1} \left(\frac{O\left(L^{3/2}R\right)}{\sqrt{m}}\right)^s m^{\frac{j-i+1}{2}} \\
\le& 0.05\sqrt{L} m^{\frac{j-i+1}{2}},
\end{aligned} \tag{19}$$

as long as $m > CL^3R^2$ which is implied by (13). This proves (17).

The proof of (15) is very similar. We still look at products of the form (18) with $j = L$. Still, there are at most $s+1$ factors of the form $W_{b:a}(0)$ each satisfying $\|W_{b:a}(0)\| \le O\left(\sqrt{L}m^{\frac{b-a+1}{2}}\right)$. However, there are at most s (instead of $s+1$) such factors such that $1 < a \le b < L$. Therefore, the product of spectral norms of all such factors is at most $\left(O(\sqrt{L})\right)^s m^{\frac{L-i+1-s}{2}}$. In other words, we can save a factor of $O(\sqrt{L})$ compared with (19). This gives us

$$\|W_{L:i}(t) - W_{L:i}(0)\| \le 0.05 m^{\frac{L-i+1}{2}},$$

proving (15).

Next we prove (16). For $1 \le i < L$, we need to bound the sum of terms of the following form:

$$W_{i:(k_s+1)}(0) \cdot \Delta_{k_s} \cdot W_{(k_s-1):(k_{s-1}+1)}(0) \cdot \Delta_{k_{s-1}} \cdots \cdot \Delta_{k_1} \cdot W_{(k_1-1):1}(0) X.$$

Again, similar as before and noting $\left\|W_{(k_1-1):1}(0)X\right\| \le \frac{5}{4}m^{\frac{k_1-1}{2}}\|X\|$, we have

$$\begin{aligned}
&\|W_{i:1}(t) - W_{i:1}(0)\| \\
\le& \sum_{s=1}^{i} \binom{j-i+1}{s} R^s \left(O\left(\sqrt{L}\right)\right)^s \cdot \frac{5}{4} m^{\frac{i-s}{2}} \|X\| \\
\le& \frac{5}{4} \sum_{s=1}^{i} L^s R^s \left(O\left(\sqrt{L}\right)\right)^s m^{\frac{i-s}{2}} \|X\| \\
=& \frac{5}{4} m^{\frac{i}{2}} \sum_{s=1}^{i} \left(\frac{O\left(L^{3/2}R\right)}{\sqrt{m}}\right)^s \|X\| \\
\le& 0.05 m^{\frac{i}{2}} \sigma_{\min}(X),
\end{aligned}$$

as long as $m > CL^3R^2 \cdot \frac{\|X\|^2}{\sigma^2_{\min}(X)} = CL^3R^2\kappa$ which is implied by (13). This finishes the proof of (16).

7.3. Proof of Claim 7.3

Recall that in Section 5 we derived (6) which is the main equation to establish convergence. In order to establish convergence from (6) we need to prove upper and lower bounds (9) on the eigenvalues of $P(t)$, as well as show that the high-order term $E(t)$ is small. We will prove these using $\mathcal{B}(t)$.

Directly from (10) and $\mathcal{B}(t)$, we have

$$\begin{aligned}
\lambda_{\max}(P(t)) &\le \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \left(\frac{5}{4}m^{\frac{i-1}{2}}\sigma_{\max}(X)\right)^2 \left(\frac{5}{4}m^{\frac{L-i}{2}}\right)^2 \\
&\le 3L\sigma^2_{\max}(X)/d_{\text{out}}, \\
\lambda_{\min}(P(t)) &\ge \frac{1}{m^{L-1}d_{\text{out}}} \sum_{i=1}^{L} \left(\frac{3}{4}m^{\frac{i-1}{2}}\sigma_{\min}(X)\right)^2 \left(\frac{3}{4}m^{\frac{L-i}{2}}\right)^2 \\
&\ge \frac{3}{10}L\sigma^2_{\min}(X)/d_{\text{out}}.
\end{aligned}$$

In Appendix F, we will prove the following bound on the high-order terms in (6):

$$\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} \|E(t)X\|_F \le \frac{1}{6}\eta\lambda_{\min}(P_t) \|U(t) - Y\|_F.$$

Finally, from (6) and $\eta \le \frac{d_{\text{out}}}{3L\|X\|^2} \le \frac{1}{\lambda_{\max}(P_t)}$ we have

$$\begin{aligned}
&\|U(t+1) - Y\|_F = \|\text{vec}\,(U(t+1) - Y)\| \\
=& \left\|(I - \eta P(t)) \cdot \text{vec}\,(U(t) - Y) + \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}\text{vec}\,(E(t)X)\right\| \\
\le& (1 - \eta\lambda_{\min}(P(t))) \|\text{vec}\,(U(t) - Y)\| + \frac{1}{\sqrt{m^{L-1}d_{\text{out}}}} \|E(t)X\|_F \\
\le& (1 - \eta\lambda_{\min}(P(t))) \|U(t) - Y\|_F + \frac{1}{6}\eta\lambda_{\min}(P_t) \|U(t) - Y\|_F \\
=& \left(1 - \frac{5}{6}\eta\lambda_{\min}(P(t))\right) \|U(t) - Y\|_F \\
=& \left(1 - \frac{1}{4}\eta L\sigma^2_{\min}(X)/d_{\text{out}}\right) \|U(t) - Y\|_F.
\end{aligned}$$

This implies $\ell(t+1) \le \left(1 - \frac{1}{4}\eta L\sigma^2_{\min}(X)/d_{\text{out}}\right)^2 \ell(t) \le \left(1 - \frac{1}{4}\eta L\sigma^2_{\min}(X)/d_{\text{out}}\right)\ell(t)$. Combined with $\mathcal{A}(t)$, this proves $\mathcal{A}(t+1)$.

8. Conclusion

We prove that gradient descent with random initialization converges to a global minimum of the ℓ_2 loss on a wide deep linear neural network. The required width in hidden layers is near linear in the depth of the network. This result improves upon previous convergence results for deep linear networks, and demonstrates that adding width can eliminate the known exponential curse of depth in linear networks. Our result may shed light on the required width in non-linear neural networks.

Acknowledgments

SSD acknowledges support from AFRL grant FA8750- 17-2-0212 and DARPA D17AP00001. WH acknowledges support from NSF, ONR, Simons Foundation, Schmidt Foundation, Mozilla Research, Amazon Research, DARPA and SRC.

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018a.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*, 2018b.

Bartlett, P., Helmbold, D., and Long, P. Gradient descent with identity initialization efficiently learns positive definite linear transformations. In *International Conference on Machine Learning*, pp. 520–529, 2018.

Brutzkus, A. and Globerson, A. Globally optimal gradient descent for a ConvNet with gaussian inputs. In *International Conference on Machine Learning*, pp. 605–614, 2017.

Du, S. S. and Lee, J. D. On the power of over-parametrization in neural networks with quadratic activation. In *International Conference on Machine Learning*, pp. 1329–1338, 2018.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *arXiv preprint arXiv:1806.00900*, 2018a.

Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018b.

Du, S. S., Lee, J. D., and Tian, Y. When is a convolutional filter easy to learn? *International Conference on Learning Representations*, 2018c.

Du, S. S., Lee, J. D., Tian, Y., Singh, A., and Poczos, B. Gradient descent learns one-hidden-layer CNN: Dont be afraid of spurious local minima. In *International Conference on Machine Learning*, pp. 1339–1348, 2018d.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eK3i09YQ.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Freeman, C. D. and Bruna, J. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pp. 797–842, 2015.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Haeffele, B. and Vidal, R. Global optimality in neural network training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4390–4398, 2017.

Hardt, M. and Ma, T. Identity matters in deep learning. *International Conference on Learning Representations*, 2016.

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

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJflg30qKX.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732, 2017.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Laurent, T. and Brecht, J. Deep linear networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning*, pp. 2908–2913, 2018.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pp. 8168–8177, 2018.

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

Lu, H. and Kawaguchi, K. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*, 2017.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. In *International Conference on Machine Learning*, pp. 2603–2612, 2017.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, 2018.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *International Conference on Learning Representations*, 2014.

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

Soltanolkotabi, M. Learning ReLUs via gradient descent. In *Advances in Neural Information Processing Systems*, pp. 2007–2017, 2017.

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

Soudry, D. and Carmon, Y. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

Tian, Y. An analytical formula of population gradient for two-layered ReLU network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

Venturi, L., Bandeira, A., and Bruna, J. Neural networks with finite intrinsic dimension have no spurious valleys. *arXiv preprint arXiv:1802.06384*, 2018.

Yun, C., Sra, S., and Jadbabaie, A. Global optimality conditions for deep neural networks. *arXiv preprint arXiv:1707.02444*, 2017.

Yun, C., Sra, S., and Jadbabaie, A. Small nonlinearities in activation functions create bad local minima in neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rke_YiRct7.

Zhang, X., Yu, Y., Wang, L., and Gu, Q. Learning one-hidden-layer ReLU networks via gradient descent. *arXiv preprint arXiv:1806.07808*, 2018.

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

Zhou, Y. and Liang, Y. Critical points of linear neural networks: Analytical forms and landscape properties. 2018.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.