SHARP CONVERGENCE ANALYSIS OF GRADIENT DESCENT FOR DEEP LINEAR NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper provides sharp rates of convergence of the gradient descent (GD) method for deep linear neural networks with different random initializations. This study touches upon one major open theoretical problem in machine learning: why deep neural networks trained with GD methods are efficient in many practical applications. While the solution of this problem is still beyond reach for general nonlinear deep neural networks, there have been extensive efforts in the literature in studying relevant questions for deep linear neural networks and there are many interesting results in this research direction. For example, recent results on the loss landscape show that even though the loss function of deep linear neural networks is non-convex, every local minimizer is also a global minimizer. When the GD method is applied to train the deep linear networks, it has been shown in the literature that the convergence behavior of the GD method depends on the initialization. In this paper, we obtain the sharp rate of convergence of GD for deep linear networks, and we show that this rate does not depend on the types of random initialization. Furthermore, we show that the depth of the network does not affect the optimal rate of convergence, provided that the width of each hidden layer is appropriately large.

1 INTRODUCTION

Deep linear neural networks, as a class of toy models, are frequently used to understand loss surfaces and the gradient-based optimization methods related to non-convex problems. The work by Dauphin et al. (2014), and Choromanska et al. (2015a) explored the loss function of deep nonlinear networks based on random matrix theory (such as spherical spin-glass model). This essentially converted the loss surface of deep nonlinear neural networks into that of deep linear neural networks under certain assumptions, some of which are unrealistic. Then, Choromanska et al. (2015b) suggested an open problem to establish a connection between the loss function of the neural networks and the Hamiltonian of the spherical spin-glass models under milder assumptions. Later, Kawaguchi (2016) successfully discarded most of these assumptions via analyzing the loss surface of the deep linear neural networks.

The landscape for deep linear neural network (Kawaguchi, 2016; Kawaguchi & Lu, 2017; Laurent & Brecht, 2018) focused on several properties of critical points: (i) every local minimum is a global minimum; (ii) every critical point that is not a local minimum is a saddle point; and (iii) there exists a saddle that all eigenvalues of its Hessian are zeros if the network is deeper than three layers. Thus, the convergence to the global minimum for deep linear neural networks suffers from the existence of poor saddles.

Lee et al. (2016) shows that the gradient method almost surely never converges to a strict saddle point, but the time cost could depend exponentially in dimension (Du et al., 2017). The GD with perturbations (Ge et al., 2015; Jin et al., 2017) can find a local minimizer in polynomial time. Thus, the trajectory approach combined with random initialization or random algorithm circumvents the obstacle of existence of poor saddles.

Studies on continuous time dynamics of gradient flow (Du et al., 2018; Arora et al., 2018c) for deep linear networks preserved balance property if the initialization is balanced. Several papers by Arora et al. (2018a;b;c), Du & Hu (2019), and Hu et al. (2020) successfully proved that the GD with corresponding initialization schemes converges to a global minimizer of the deep linear neural

networks with high probability. Furthermore, the rate of convergence is linear, and it behaves like GD for a convex problem.

Hu et al. (2020) established that the convergence for Gaussian initialization could be very slow for large depth deep linear neural networks, unless the width is almost linear in depth. They also show that the orthogonal initialization in deep linear neural networks will speed up convergence. Thus, the convergence behavior of the GD method to train deep linear neural networks depends crucially on the initialization.

Recent discoveries found the connection between deep learning and kernel methods (Daniely, 2017; Arora et al., 2019a;b; Chizat et al., 2019; Lee et al., 2019; Du et al., 2019; Cao & Gu, 2019; Woodworth et al., 2020), especially the neural tangent kernel (NTK), introduced by Jacot et al. (2018). For most common neural networks, the NTK becomes constant (Jacot et al., 2018; Liu et al., 2020), and the NTK remains constant throughout training in the limit of large layer width. The neural networks are well described throughout training by their first order Taylor expansion around their parameters at initialization (Lee et al., 2019).

In this paper, we first study the convergence region, the set of initialization which lead to the linear convergence of the GD for the deep linear neural networks (Lemma D.1). Next, we demonstrate that if the minimum width among all hidden layers is sufficiently large, then the random initialization will fall into the convergence region with high probability (Theorem 3.1, Theorem 3.2, Theorem B.1, Theorem B.2, and Theorem B.3). Furthermore, the worst-case convergence rate of the GD for the deep linear neural networks is almost the same as the original convex problem with a corresponding learning rate. We also demonstrate that the trajectories of the GD for the deep linear neural networks are arbitrarily close to those for the convex problem. The precise statement is related to remark 4, Theorem 3.3, Lemma D.5 and Corollary 1.

This work is inspired by recent work (Du & Hu, 2019; Hu et al., 2020), which carefully constructed the upper and lower bounds of eigenvalues of this Gram matrix along the GD and established linear convergence. In this paper, we generalize their results to strongly convex loss functions with layer-varying width and obtain sharper results. We also show that, our rate of convergence for GD in deep linear neural networks is sharp in the sense that it matches the worst-case convergence rate for the original convex problem. The trajectories between the GD for deep linear neural networks and the original convex problem (1) can be arbitrary close. Furthermore we show that the optimal rate neither depends on the types of random initialization nor the depth of the network, provided that the width of each hidden layer is appropriately large.

2 Preliminaries

First let's start with some notations to set up the problem.

2.1 Problem setup

Let $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$ be an input vector and a target vector, respectively. Define $\{(x_i, y_i)\}_{i=1}^m$ as a training dataset of size m, and let $X = [x_1, x_2, \cdots, x_m] \neq 0$ and $Y = [y_1, y_2, \cdots, y_m]$. Denote the weight parameters by $W \in \mathbb{R}^{n_y \times n_x}$.

Consider the well studied convex optimization problem:

$$\underset{W}{\text{minimize}} \quad L(W) := \frac{1}{m} \sum_{i=1}^{m} l(Wx_i, y_i). \tag{1}$$

The GD for convex problem (1) with learning rate η_* is given by:

$$W(t+1) = W(t) - \eta_* \nabla L(W(t)), t = 0, 1, 2, \cdots. \tag{2}$$

For any matrix A, let $\sigma_{max}(A)$ and $\sigma_{min}(A)$ be the largest and smallest singular values of A respectively. In this paper, we consider two types of matrix norms and one type of semi-norm for A,

$$\|A\| := \sigma_{max}(A), \|A\|_F^2 := tr(AA^T), \text{ and } \|A\|_X := \|AP_X\|_F,$$

where $P_X = X(X^TX)^\dagger X^T$ is the orthogonal projection matrix onto the column space of X, and $(X^TX)^\dagger$ is the Moore–Penrose inverse.

For two real matrices A, B with the same size, we consider their Frobenius inner product as well as their semi-inner product,

$$\langle A, B\rangle = \langle A, B\rangle_F := tr(A^TB),$$
$$\langle A, B\rangle_X := \langle AP_X, BP_X\rangle.$$

We list some basic properties for the semi-norm and semi-inner product.

Lemma 2.1. *The loss function $L(W)$ defined in (1) satisfies the following properties: for any $W, V \in \mathbb{R}^{n_y\times n_x}$,*

1. $L(W) = L(WP_X)$,
2. $\nabla L(W) = \nabla L(WP_X)P_X$,
3. $\langle \nabla L(W), V\rangle_F = \langle \nabla L(W), V\rangle_X$,
4. $\|\nabla L(W)\|_F = \|\nabla L(W)\|_X$,
5. $\|W\|_X \equiv \|W\|_F$ *if and only if X is full row rank.*

The next lemma illustrates why semi-norm $\|\cdot\|_X$ is crucial in our analysis.

Lemma 2.2. *Assume that $l(\cdot, y)$ is $\alpha(l)-$strongly convex. The following statements hold.*

1. *If X is not full row rank, then $L(W)$ is neither strictly convex nor strongly convex with respect to $\|\cdot\|_F$.*
2. *$L(W)$ is $\frac{\alpha(l)\lambda_{min}(XX^T)}{m}-$strongly convex with respect to $\|\cdot\|_X$, where $\lambda_{min}(XX^T)$ is the smallest non-zero eigenvalue of XX^T.*

The proofs of the two lemmas above can be found in appendix A. From now on, if we do not specify the inner product, we will always consider the semi-inner product.

Assume that L is $\alpha-$strongly convex ($\alpha > 0$), and ∇L is $\beta-$Lipschitz (with respect to semi-norm $\|\cdot\|_X$), that means, for any $W, V \in \mathbb{R}^{n_y\times n_x}$,

$$L(W) \geq L(V) + \langle \nabla L(V), W - V\rangle_X + \frac{\alpha}{2}\|W - V\|_X^2,$$
$$\|\nabla L(W) - \nabla L(V)\|_X = \|\nabla L(W) - \nabla L(V)\|_F \leq \beta\|W - V\|_X.$$

Without loss of generality, we assume α and β are the best constants. Then, Lemma 2.2 implies $\alpha \geq \frac{\alpha(l)\lambda_{min}(XX^T)}{m}$. Similarly, we can also show that $\beta \leq \frac{\beta(l)\lambda_{max}(XX^T)}{m}$, where $\nabla l(\cdot, y)$ is $\beta(l)-$Lipschitz and $\lambda_{max}(XX^T)$ is the largest eigenvalue of XX^T.

Define the effective condition number of the convex function L by $\kappa = \kappa(L) = \frac{\beta}{\alpha} < \infty$, κ appears naturally in the rate of convergence of GD. Let W_* be a global minimizer of $L(W)$, that is $L(W_*) = \min_W L(W)$. Notice that W_* might not be unique, but W_*P_X is unique.

The well known results on rate of convergence of GD (2) state

$$\eta_* = \frac{1}{\beta} \implies \mathcal{E}(t) \leq \left(1 - \frac{1}{\kappa}\right)^t \mathcal{E}(0), t = 1, 2, \cdots, \tag{3}$$

as well as

$$\eta_* = \frac{2}{\alpha + \beta} \implies \mathcal{E}(t) \leq \left(1 - \frac{4\kappa}{(1+\kappa)^2}\right)^t \|W(0) - W_*\|_X^2, t = 1, 2, \cdots, \tag{4}$$

where $\mathcal{E}(t) = L(W(t)) - L(W_*)$.

2.2 DEEP LINEAR NETWORK SETUP

Let $N-1$ be the number of hidden layers. Assume $rank(X)=r$. Denote the weight parameters by $W_k \in \mathbb{R}^{n_k \times n_{k-1}}, k=1,2\cdots,N$, with $n_N=n_y, n_0=n_x$, where the n_k is the width of the $k-th$ layer. Set $n_{min}=\min\{n_1,n_2,\cdots,n_{N-1}\}$, and $n_{max}=\max\{n_1,n_2,\cdots,n_{N-1}\}$. For notational convenience, we denote $n_{j:i}=\prod_{i\le k\le j} n_k$ and denote $W_{j:i}=W_jW_{j-1}\cdots W_i$ for each $1\le i\le j\le N$. Define $n_{i-1:i}=1$ and $W_{i-1:i}=I$ (of appropriate dimension) for completeness.

Consider the implicit regularization $W=W_{N:1}$ for the convex problem (1). Thus, we obtain the following non-convex optimization problem of deep linear neural networks:

$$\underset{W_1,\cdots,W_N}{\text{minimize}} \quad L(W_{N:1})=\frac{1}{m}\sum_{i=1}^{m} l(W_{N:1}x_i,y_i). \tag{5}$$

Example 2.1. Specifically, if we set the loss to be $l(Wx_i,y_i)=\|Wx_i-y_i\|_2^2$, then $L(W)=\frac{1}{m}\|WX-Y\|_F^2$ is $\frac{2\lambda_{min}(XX^T)}{m}$−strongly convex, and ∇L is $\frac{2\lambda_{max}(XX^T)}{m}$−Lipschitz.

Example 2.2. The deep linear neural networks with regularization $\lambda\|W_N\cdots W_1P_X\|_F^2$ could be converted into a new optimization problem

$$\underset{W_1,\cdots,W_N}{\text{minimize}} \quad L(W_{N:1})+\lambda\|W_{N:1}\|_X^2 .$$

Let $L_\lambda(W)=L(W)+\lambda\|W\|_X^2$. Then $L_\lambda(\cdot)$ is $\alpha+2\lambda$-strongly convex, and $\nabla L_\lambda(\cdot)$ is $\beta+2\lambda$-Lipschitz.

More general, if we consider regularization with form $R(W)=\lambda\cdot g(WP_X)$, and $g(\cdot)$ is α'-strongly convex, and β'-Lipschitz, then for the optimization problem

$$\underset{W_1,\cdots,W_N}{\text{minimize}} \quad L(W_{N:1})+R(W_N\cdots W_1)=: L_R(W_N\cdots W_1),$$

we know that $L_R(\cdot)$ is $\alpha+\lambda\alpha'$-strongly convex, and $\nabla L_R(\cdot)$ is $\beta+\lambda\beta'$-Lipschitz.

2.3 INITIALIZATION SCHEMES

In literature, researchers consider the following form of deep linear network instead of (5):

$$\underset{W_1,\cdots,W_N}{\text{minimize}} \quad L(a_NW_{N:1})=\frac{1}{m}\sum_{i=1}^{m} l(a_NW_{N:1}x_i,y_i), \tag{6}$$

where $a_N=1/\sqrt{n_1n_2\cdots n_N}$ is a normalization constant.

Applying GD on (6), where we update W_j simultaneously for j,

$$W_j(t+1)=W_j(t)-\eta\cdot a_N\left(W_{N:j+1}(t)\right)^T\nabla L\left(a_NW_{N:1}(t)\right)\left(W_{j-1:1}(t)\right)^T, j=1,\cdots,N. \tag{7}$$

Recent work consider GD (7), initializing $W_j(0)$ with Gaussian initialization (Du & Hu, 2019) or scaled orthogonal initialization (Hu et al., 2020).

In this paper, we consider the following three kinds of random initialization which generalizes their idea.

Gaussian initialization

Let $W_1(0),\cdots,W_N(0)$ be the weight matrices at initialization. We assume that all entries of $W_j, 1\le j\le N$ are independent Gaussian random variables with zero mean and unit variance.

Then a_N is a normalization constant in the sense that for any $x\in\mathbb{R}^{n_0}$, we have

$$\mathbb{E}\left[\|a_NW_{N:1}(0)x\|_2^2\right]=\|x\|_2^2 . \tag{8}$$

In fact, all initialization we discussed in this paper satisfy (8).

Remark 1. Let $V_i(t) = \frac{1}{\sqrt{n_i}} W_i(t)$, for $1 \leq i \leq N$. Then, the GD (7) with unit variance Gaussian initialization is equivalent to

$$V_j(t+1) = V_j(t) - \frac{\eta}{n_j} \left(V_{N:j+1}(t)\right)^T \nabla L\left(V_{N:1}(t)\right) \left(V_{j-1:1}(t)\right)^T, \tag{9}$$

with mean zero and variance $\frac{1}{n_i}$ Gaussian initialization.

The GD (9) for the loss (5) is equivalent to the GD (7) for the loss (6). From now on, we will only consider GD (7) for deep linear neural networks (6).

Orthogonal initialization

We consider a so-called one peak random orthogonal projections and embeddings initialization, which generalizes the idea of orthogonal initialization (Hu et al., 2020).

Definition 2.1. A initialization $W_{N:1}(0) = W_N(0)W_{N-1}(0)\cdots W_1(0)$ is said to be one peak random orthogonal projections and embeddings initialization if there exists $1 \leq p < N$, such that $n_0 \leq n_1 \leq n_2 \leq \cdots \leq n_p$, $n_p \geq n_{p+1} \geq n_{p+2} \geq \cdots n_{N-1} \geq n_N$, and $W_1(0), W_2(0), \cdots, W_p(0), W_{p+1}(0), W_{p+2}(0), \cdots, W_{n_N}(0)$ are independent and uniformly distributed over rectangular matrices which satisfy

$$\begin{cases} W_i^T(0)W_i(0) = n_i I_{n_i}, 1 \leq i \leq p, \\ W_j(0)W_j^T(0) = n_{j-1} I_{n_{j-1}}, p+1 \leq j \leq N. \end{cases}$$

Remark 2. In this definition, $\sqrt{\frac{1}{n_i}} W_i(0), 1 \leq i \leq p$ are random embeddings and $\sqrt{\frac{1}{n_{j-1}}} W_j(0), p+1 \leq j \leq N$ are random orthogonal projections. Notice that, A is a random orthogonal projection if and only if A^T is a random embedding.

Remark 3. There are several ways to construct random matrix $A = (a_{ij})_{q\times n},\ q \leq n$, which is uniformly distributed over rectangular matrices with $AA^T = c^2 I_q, c > 0$. Let O_n be uniformly distributed over real orthogonal group of order n, and O_n is partitioned as $O_n = (A_1^T, A_2^T)^T$, where A_1 is $q \times n$. Assume $X = (x_{ij})_{q\times n}$, and x_{ij} are independent standard normal random variables. Then A, cA_1, and $c(XX^T)^{-1/2}X$ have the same distribution. More detailed properties are listed in Proposition F.1.

Arora et al. (2018a) studied the rate of convergence to global optimum for GD to train deep linear neural network for balanced initialization. Here we will consider a special case of balanced initialization which is described as follows.

Special balanced initialization

Assume $n_1 = \cdots = n_{N-1} = n$. Consider initialization $W_N(0) = \sqrt{n} U_N [I_{n_y}, 0_{n_y \times (n-n_y)}] V_N^T$, $W_1(0) = \sqrt{n} U_1 [I_{n_x}, 0_{n_x \times (n-n_x)}]^T V_1^T$ and $W_i(0) = \sqrt{n} U_i I_n V_i^T, 2 \leq i \leq N-1$, where U_N, U_{N-1}, $V_i = U_{i-1}, 2 \leq i \leq N-1$ are orthogonal matrices (random or deterministic), and V_N has uniform distribution over orthogonal matrices.

A simple estimation of loss at the initialization is given by the following lemma.

Lemma 2.3. *If the initialization satisfies (8) for all x, then with probability at least $1 - \frac{\delta}{2}$, we have*

$$L(a_N W_{N:1}(0)) - L(W_*) \leq \beta B_\delta,$$

where $B_\delta = \left(\frac{2 \cdot rank(X)}{\delta} + \|W_*\|_X^2\right)$.

Note that using sharp concentration inequality, the bound B_δ could be improved.

3 Main theorems

Assume the thinnest layer is either the input layer or the output layer, that is $n_{min} \geq \max\{n_0, n_N\}$ and the ratio between the width of any hidden layer is bounded from above, precisely we have $\frac{n_{max}}{n_{min}} \leq C_0 < \infty$. The quantities $C_1, \cdots, C_5$ and q are defined at the beginning of appendix D.

For notational convenience, we denote

$$\mathcal{E}(t) = L(W(t)) - L(W_*), \text{ and } \mathcal{E}_{DLN}(t) = L(a_N W_{N:1}(t)) - L(W_*).$$

Our assumptions and notation are now in place, and we next state our main theorems in this section.

3.1 Linear convergence of Deep linear neural networks

In appendix B, we present a sharp estimate on the linear convergence of the GD for deep linear neural networks, in Theorem B.1 for Gaussian initialization, Theorem B.2 for orthogonal initialization, and Theorem B.3 for special balanced initialization, respectively. In particular, with the special choice of learning rate $\eta = \frac{n_N}{\beta N}$, Theorem B.1 and Theorem B.2 lead us to obtain the following optimum rate of convergence:

Theorem 3.1. *Given any $\delta, \varepsilon \in (0, \frac{1}{2})$, there exists a constant $C := C(\varepsilon)$, such that if overparameterization condition $n_{min} \geq C \cdot C_2 \cdot N$ holds, then with probability at least $1 - \delta$ over the Gaussian initialization, we have*

$$\mathcal{E}_{DLN}(t) \leq \left(1 - \frac{1-\varepsilon}{\kappa}\right)^t \mathcal{E}_{DLN}(0), t = 1, 2, \cdots .$$

Theorem 3.2. *Given any $\delta, \varepsilon \in (0, \frac{1}{2})$, there exists a constant $C := C(\varepsilon)$, such that if overparameterization condition $n_{min} \geq C \cdot C_5$ holds, then with probability at least $1 - \delta$ over the one peak random projections and embeddings initialization, we have*

$$\mathcal{E}_{DLN}(t) \leq \left(1 - \frac{1-\varepsilon}{\kappa}\right)^t \mathcal{E}_{DLN}(0), t = 1, 2, \cdots .$$

Remark 4. Consider the GD (2) with learning rate $\eta_* = \frac{1}{\beta}$, and initialization $W(0) = a_N W_{N:1}(0)$. The well known result on the rate of convergence (3) for the GD (2) of convex problem (1) matches the rates in Theorem B.1 and Theorem B.2.

3.2 Results on trajectories

Theorem 3.1 and Theorem 3.2 combined with remark 4 establish that the rate of convergence to a global optimum for GD to train a deep linear neural networks is almost the same as that to train the corresponding convex problem with arbitrarily high probability, if the width is large enough.

Moreover, the GD for the fully-connected deep linear neural networks (7) and that for the GD (2) have almost the same trajectories.

Let $\eta_1 = \frac{2n_N}{\beta N}$ be an upper bound of learning rate η. We can show that the trajectories of GD (7) for deep linear neural networks (6) with learing rate $\eta < \eta_1$ are close to those of GD (2) with learning rate $\eta_* = \frac{N}{n_N}\eta$ for the corresponding convex problem (1) with high probability, if the width of each hidden layer is sufficiently large. The precise statement is as follows.

Theorem 3.3. *Consider the GD for the deep linear neural networks (7) with learning rate $\eta < \eta_1$ for $a_N W_{N:1}(t)$, $t = 0, 1, \cdots$, and the GD (2) with learning rate $\eta_* = \frac{N}{n_N}\eta$ for $W(t)$, $t = 0, 1, \cdots$. Given $\tau, \delta \in (0, 1)$, there exists a constant $C := C(\tau, \eta/\eta_1)$ such that if one of the following three overparameterization conditions holds:*

1. *$n_{min} \geq C \cdot C_2 \cdot N$ with the Gaussian initialization,*
2. *$n_{min} \geq C \cdot C_5$ with the one peak random orthogonal projections and embeddings initialization,*
3. *$n_{min} \geq C \cdot C_6$ with the special balanced initialization,*

then with probability at least $1 - \delta$, we have

$$\|a_N W_{N:1}(t) - W(t)\|_X^2 \leq D(\tau, q, t) \|a_N W_{N:1}(0) - W_*\|_X^2 , \tag{10a}$$

$$|\mathcal{E}_{DLN}(t) - \mathcal{E}(t)| \leq \beta \left(q^{t/2}\sqrt{D(\tau, q, t)} + \frac{1}{2}D(\tau, q, t)\right) \|a_N W_{N:1}(0) - W_*\|_X^2 , \tag{10b}$$

$$\mathcal{E}_{DLN}(t) \leq 3\beta(q + \tau)^t \|a_N W_{N:1}(0) - W_*\|_X^2 , \tag{10c}$$

where $D(\tau, q, t) = \min\left\{\frac{\tau}{1-q}, 2(q+\tau)^t\right\}$*, with* q *defined in (13).*

Corollary 1. *Under the setting of Theorem 3.3, if we set* $\eta = \frac{2n_N}{(\alpha+\beta)N}$*, the following inequality hold with high probability,*

$$\mathcal{E}_{DLN}(t) \leq 3\beta\left(1 - \frac{4\kappa}{(1+\kappa)^2} + \tau\right)^t \|a_N W_{N:1}(0) - W_*\|_X^2 . \tag{11}$$

We notice that the rate of convergence in (11) is better than that in Theorem 3.1 and Theorem 3.2. It is because if $\kappa > 1$*, we can choose* τ *sufficiently small, so that the following inequality holds:*

$$1 - \frac{4\kappa}{(1+\kappa)^2} + \tau < 1 - \frac{1}{\kappa}.$$

Roughly speaking, Theorem 3.1, Theorem 3.2, Theorem 3.3, Theorem B.1, together with Theorem B.2 illustrate that the implicit regularization induced by GD for a convex problem recovers the convex problem itself in terms of optimization, at the cost of linear convergence only with high probability for random initialization.

4 Experiments

In this section, we provide some empirical evidence to support the results in Section 3. We will show how the trajectories of the non-convex deep linear neural networks are related to a convex optimization problem for GD under different initialization schemes.

a) We choose $X \in \mathbb{R}^{128\times 1000}$ and $W_* \in \mathbb{R}^{10\times 128}$ and set $Y = W_* + \varepsilon$, where the entries in X, W_* and ε are drawn i.i.d. from $N(0,1)$.

b) We consider the loss function $\frac{1}{2}\|a_N W_{N:1} - Y\|_F^2$.

c) For the given linear networks, we apply the Gaussian initialization and the one peak random orthogonal projections and embeddings initialization, which are denoted as $W_j(0), 1 \leq j \leq N$.

d) For the convex optimization problem (1), we set the initialization to be $W(0) = a_N W_N(0) \cdots W_1(0)$.

d) We set the learning rate $\eta = \frac{n_N}{N\cdot\|X\|^2}$ and $\eta_* = \frac{N}{n_N}\eta$ for the deep linear neural networks and the convex problem, separately.

e) We draw the loss function through 25 iterations.

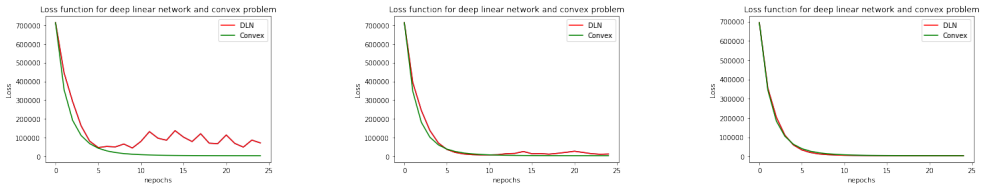

Figure 1: Plot of Loss as a function of number of iterations with $n_1 = n_2 = n_3 = 128$ (Left), 200 (Middle), 300 (Right) for Gaussian initialization, respectively.

Figure 1 and 2 show plot of loss as a function of number of iterations. When n is small, the trajectories of loss for deep linear neural networks are not even decreasing in some iterations; when n is large, the trajectories of loss for deep linear neural networks are close to those for the corresponding convex problem.

5 Overview of the Proofs of Main Theorems and Lemmas

Now we will give an overview of the proofs for all theorems in the main results. Since Theorem 3.1 and Theorem 3.2 in the main results are special cases of Theorem B.1 and Theorem B.2, we only need to focus on the proofs of Theorem B.1, Theorem B.2, Theorem B.3, and Theorem 3.3.

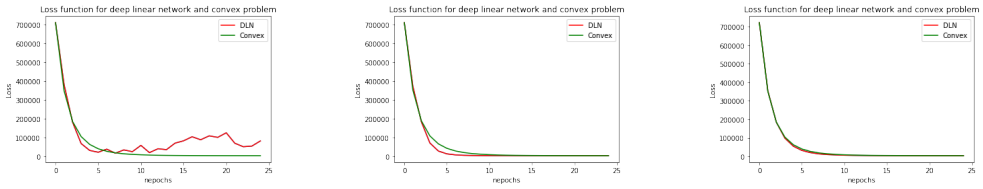

Figure 2: Plot of Loss as a function of number of iterations with $n_1 = n_2 = n_3 = 128$ (Left), 200 (Middle), 2000 (Right) for Orthogonal initialization, respectively.

We start with the convergence region of deep linear neural networks. Basically, the convergence region is the set of initialization which lead to the convergence of the GD for the deep linear neural networks. The precise definition can be found in the appendix D. Lemma D.1 and Lemma D.5 prove that this convergence region satisfies the following properties: if the initialization falls into the convergence region, then

(i) the GD is guaranteed to converge to a global minimizer of the deep linear neural networks,

(ii) the worst-case rate of convergence of the GD for the deep linear neural networks, which is a non-convex problem, is almost the same as the corresponding convex problem with a corresponding learning rate, and,

(iii) the trajectories of the GD for the deep linear neural networks are arbitrarily close to those for the corresponding convex problem.

More precisely, Lemma D.1 establishes the convergence region for deterministic initialization, and it demonstrates the first two properties, (i) and (ii). Additionally, in appendix E and appendix F we also proved that the spectral properties of products of random matrices partially reveal the mystery of overparameterization, that is, overparameterization by adding width of each hidden layer guarantees that the random initialization will fall into the convergence region with high probability. These results provide a foundation to establish Theorem B.1, B.2, and B.3.

On the other hand, Lemma D.3 shows that if the initialization falls into the convergence region, the update rule for the product of weight matrices in GD for deep linear neural networks is more or less that given in (2). This result can be used to establish both Lemma D.5, and Theorem 3.3, which is precisely the property (iii) of the convergence region for deterministic initialization and non-deterministic initialization, respectively.

REFERENCES

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *CoRR*, abs/1810.02281, 2018a.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *CoRR*, abs/1802.06509, 2018b.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pp. 244–253. PMLR, 2018c.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019b.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Djalil Chafaı, Djalil Chafä, Olivier Guédon, Guillaume Lecue, and Alain Pajor. Singular values of random matrices. *Lecture Notes*, 2009.

Lénaïc Chizat, Edouard Oyallon, and Francis R. Bach. On lazy training in differentiable programming. In *NeurIPS*, 2019.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204. PMLR, 2015a.

Anna Choromanska, Yann LeCun, and Gérard Ben Arous. Open problem: The landscape of the loss surfaces of multilayer networks. In *Conference on Learning Theory*, pp. 1756–1760. PMLR, 2015b.

Amit Daniely. Sgd learns the conjugate kernel class of the network. *Advances in Neural Information Processing Systems*, 30, 2017.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 27, 2014.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pp. 1675–1685. PMLR, 2019.

Simon S. Du and Wei Hu. Width provably matters in optimization for deep linear neural networks, 2019.

Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Aarti Singh, and Barnabas Poczos. Gradient descent can take exponential time to escape saddle points. *Advances in neural information processing systems*, 30, 2017.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in Neural Information Processing Systems*, 31, 2018.

Morris L. Eaton. Group invariance applications in statistics. *Regional Conference Series in Probability and Statistics*, 1:i–133, 1989. ISSN 19355912.

Sam Elder. Bayesian adaptive data analysis guarantees from subgaussianity. *CoRR*, abs/1611.00065, 2016.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pp. 797–842. PMLR, 2015.

Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

GJO Jameson. Inequalities for gamma function ratios. *The American Mathematical Monthly*, 120 (10):936–940, 2013.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

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

Kenji Kawaguchi and Haihao Lu. Deep creates no bad local minima. *arXiv: 1702.08580*, 2017.

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

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on learning theory*, pp. 1246–1257. PMLR, 2016.

Chaoyue Liu, Libin Zhu, and Misha Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33: 15954–15964, 2020.

Olivier Marchal and Julyan Arbel. On the sub-gaussianity of the beta and dirichlet distributions. *Electronic Communications in Probability*, 22(none):1 – 14, 2017.

Feng Qi, Bai-Ni Guo, and C. Chen. The best bounds in gautschi-kershaw inequalities. *Mathematical Inequalities Applications*, pp. 427–436, 2006.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.