Faster Directional Convergence of Linear Neural Networks under Spherically Symmetric Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we study (deep) linear neural networks with the logit loss and gradient methods on the zero-margin separable data, which is not covered by previous works. In particular, we show directional convergence guarantees with a superpolynomial convergence rate for (deep) linear networks under the assumption of spherically symmetric data distribution, which can be viewed as a specific zero-margin dataset. Furthermore, our results are built on the dynamic without small initial loss or presumed convergence of weight direction, or overparameterization constraint in contrast to previous works. We also characterize our findings from benign data case to real dataset and non-linear networks particularly in early training phase.

1 Introduction

In recent years, deep neural networks have been successfully trained with simple gradient-based methods, despite the inherent non-convexity of the learning problem. Meanwhile, implicit biases introduced by optimization algorithms play a crucial role in training neural networks. Previous work rigorously analyzes the optimization of deep networks, leading to many exciting developments such as the neural tangent kernel [1, 4, 12, 13, 20, 30]. The above works are based on over-parameterization regime, which makes the weights stay close to initialization and share similar dynamic with linear regression. By contrast, Lyu and Li [32] showed that if the data can be perfectly classified, the parameters are guaranteed to diverge in norm to infinity. Moreover, Ji and Telgarsky [22] experimentally illustrated that the prediction surface continues to change even with large width. These findings raise an issue that the properties about neural networks may be unstable if the prediction surface never stops changing.

Lyu and Li [32], Ji and Telgarsky [22, 24] already addressed this issue by guaranteeing stable (directional) convergence behavior of deep networks as training proceeds, despite the growth of weight vectors to infinity. Their works focused on general homogeneous deep networks and exponential-type loss functions at the "late training" phase, meaning that the predictor has already obtained zero classification error. Moreover, they proved asymptotic directional convergence, i.e., the parameters converge in direction, but had no exact directional convergence rate. And we would like to understand the whole training dynamic through rigorous analysis on more specific setting.

In addition, the implicit bias of gradient methods has been extensively studied recently. Specifically, for linear classifiers on separable data and with exponentially-tailed losses, gradient descent converges to the ℓ_2 maximum margin direction [23, 33, 34, 42, 43]. However, these work consider a classical finite data with positive margin. Ji and Telgarsky [21], Ji et al. [25] considered a non-separable case, but requiring a positive margin in the "separable part". And if the margin becomes particularly small, previous results might become vacuous and the training period might last for certainly long due to the inverse ratio with such a margin. Moreover, we may encounter large-scale dataset, which is typically

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

inseparable or separable with particularly small margin. Thus, understanding the zero-margin scheme could be seen as a completion of the setting.

In this paper, we consider a specific case for binary classification with zero margin under population logit loss (binary cross-entropy). To simulate infinite data case and obtain simplified analysis, we make the assumption that the distribution of input data is spherically symmetric, which includes the standard Gaussian and uniform distribution on the spherical surface. The main contributions of this paper are summarized as follows:

- We present two-phase directional convergence of linear classifier under gradient methods. Particularly, we show the descending and ascending behavior of weight norm, leading super-polynomial directional convergence correspondingly. Our findings improve convergence bound compared to the general empirical loss and positive margin setting as we expected, because we make the benign distribution assumption.
- Invoking from linear predictor and previous work, we also build up (at least) two-phase directional convergence rates for deep linear networks. In particular, the induced weight norm still goes through descending and ascending period. The derivation of our result is concise and does not rely on the assumption of zero classification error or overparameterization.
- We also verify our results in numerical experiments and show potential improvements of our results. Moreover, we observe simple possible accelerated methods from our theoretical findings, through increasing data norm, or induced weight norm, or depth of networks. Finally, we also observe the descending and ascending behavior of weight norm in real datasets and non-linear networks.

Note that previous works also make assumptions [44, 18], but some may not explain practical experiments, such as the descent-ascent variation of weight norm from our more tight analysis, and this is the fundamental trend of understanding neural networks through optimization trajectory.

1.1 Related work

There is a rapid growth of literature, surveying all of which well is outside our scope. Thus, we only briefly review the works most related to ours.

Optimization dynamics of linear neural networks. A large amount of works focus on training networks under regression case with the square loss. Yehudai and Ohad [45] analyzed a single neuron in a realizable setting and prove linear convergence under mild assumptions. As for linear networks, Saxe et al. [39] analyzed the trajectory of parameters under spectral initialization. Bartlett et al. [6], Arora et al. [2], Hu et al. [19], Du and Hu [11] showed linear convergence provided that different conditions hold, such as identity, balanced, orthogonal initialization, or suitable overparameterization. Eftekhari [14] broke away from "lazy" training regime introduced by overparameterization and provides non-local convergence under the scalar output. There also has plenty of literature employing cross entropy loss under classification tasks. Several works [8, 9, 30] employ small variation of activation patterns or weights near the initialization under overparameterized neural networks with cross entropy loss. Shamir [41] showed the number of iterations required for convergence scales exponentially with the depth in deep linear networks, leading to the necessity of good initialization.

Directional convergence. Directional convergence has been established for linear predictors [10, 16, 17, 21, 25, 33, 34, 42, 43], coupled with separated data [5, 15, 40]. These works consider linear classifiers with smooth monotone loss functions including the cross-entropy loss, optimized on linearly separable (or nonseparable) data similar to us but with finite data as realistic setting. Moreover, Ji and Telgarsky [23] showed tight results in training samples through the optimal solution of a dual optimization problem given by a smoothed margin, even for general losses. Lyu and Li [32], Ji and Telgarsky [22, 24] extended linear classifiers to deep homogeneous networks using powerful techniques. The findings is built on the alignment of some weights of neural networks reaching a stationary point of the limiting margin maximization objective under gradient methods. Chizat and Bach [10] improved such a result for the two-layer case, identifying the learned classifier as the solution of a convex max-margin problem. Yun et al. [46] provided a unified framework that connects multiple existing results on implicit bias of gradient flow under a general tensor formulation of linear networks.

2 Preliminaries

In this paper, we would learn predictor $\phi(\cdot, \mathbf{w})\colon \mathbb{R}^d \to \mathbb{R}$ with parameters $\mathbf{w}$, and let $\mathbf{x} \in \mathbb{R}^d$ be input features sampled from distribution $\mathcal{D}$ with a well-defined covariance matrix. We consider a binary classification problem in which the label for each $\mathbf{x}$ is decided by such a normalized vector $\mathbf{v} \in \mathbb{R}^d$ that $\|\mathbf{v}\| = 1$, and in this way $y(\mathbf{x}) = \mathrm{sgn}(\mathbf{v}^\top \mathbf{x}) \in \{-1, +1\}$ [1]. We employ the population logit loss $\ell(z) := \ln(1 + e^{-z})$ (binary cross-entropy) with $z(\mathbf{x}, \mathbf{w}) = y(\mathbf{x}) \cdot \phi(\mathbf{x}, \mathbf{w})$. Thus, the objective is

$$\min_{\mathbf{w}} L(\mathbf{w}) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \ln\left(1 + e^{-y(\mathbf{x})\phi(\mathbf{x},\mathbf{w})}\right). \tag{1}$$

We focus on the following standard gradient methods applied with $\nabla L(\mathbf{w})$:

1) **Gradient flow**: We initialize $\mathbf{w}(0)$, and for every $t > 0$ let $\mathbf{w}(t)$ be the solution of the differential equation: $\dot{\mathbf{w}}(t) = -\nabla L(\mathbf{w}(t))$.

2) **Gradient descent**: We initialize $\mathbf{w}(0)$ and set a sequence of positive learning rates $\{\eta_n\}_{n=1}^{\infty}$. At each iteration $t > 0$, we do a single step in the negative direction of the gradient, that is, $\mathbf{w}(n+1) = \mathbf{w}(n) - \eta_n \nabla L(\mathbf{w}(n))$.

Notation. In this paper, $\|\cdot\|$ denotes the standard ℓ_2-norm, $\Sigma = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \mathbf{x}\mathbf{x}^\top$ is the population covariance matrix, $\lambda_{\max}(A)$ is the maximum eigenvalue of a real symmetric matrix A and $\mathcal{S}^{d-1}$ is the surface of an d-dimensional unit sphere.

We let $\overline{\mathbf{w}} := \mathbf{w}/\|\mathbf{w}\|$ whenever $\|\mathbf{w}\| \neq 0$. Moreover, we denote a vector $\mathbf{w} = (w_1, \ldots, w_d)^\top \in \mathbb{R}^d$. Given vectors $\mathbf{w}$ and $\mathbf{v}$, we let $\theta(\mathbf{w}, \mathbf{v}) := \arccos\left[\mathbf{w}^\top \mathbf{v}/\left(\|\mathbf{w}\| \cdot \|\mathbf{v}\|\right)\right] \in [0, \pi]$ denote the angle between $\mathbf{w}$ and $\mathbf{v}$, $\theta(t) := \theta(\mathbf{w}(t), \mathbf{v})$ and $\theta(n) := \theta(\mathbf{w}(n), \mathbf{v})$. We let $\mathcal{D}_{\mathbf{w},\mathbf{v}}$ be the marginal distribution of $\mathbf{x}$ on the subspace spanned by two linearly independent vectors $\mathbf{w}, \mathbf{v}$ (a distribution over $\mathbb{R}^2$), $\mathcal{D}_2 := \mathcal{D}_{\mathbf{e}_1,\mathbf{e}_2}$ where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ are the d-dimensional coordinate directions, and $c_0 := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_2}\|\mathbf{x}\|$. Additionally, we call complexity $\mathcal{O}\left(\ln^{\alpha}(1/\epsilon)\right)$ for a constant $\alpha > 0$ to obtain ϵ-error as *superpolynomial* convergence.

2.1 Basic properties

To build the rough understanding of such an objective function, we give the following properties.

Proposition 1 *The objective function $L(\mathbf{w})$ is c-Lipschitz continuous, $\frac{1}{4}\lambda_{\max}(\Sigma)$-smooth, and convex but not strongly convex if $P\{\mathbf{x} : \mathbf{v}^\top \mathbf{x} = 0\} = 0$. Here $c := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \|\mathbf{x}\|$.*

Generally speaking, gradient methods have no linear convergence for smooth, Lipschitz continuous, and convex but not strongly convex functions [35]. However, some invariant properties (such as Proposition 2 below) make the training dynamic reserve linear directional convergence.

Proposition 2 *Suppose $P\left(\{\mathbf{x} : \mathbf{v}^\top \mathbf{x} = 0\}\right) < 1$. Then $\mathbf{v}^\top \nabla L(\mathbf{w}) < 0$ for any $\mathbf{w} \in \mathbb{R}^d$. Therefore, $\mathbf{v}^\top \mathbf{w}(t)$ is increasing, and $\|\mathbf{w}(t)\|$ is unbounded for gradient flow. If the learning rates are lower bounded, i.e. $\eta_n \geq \eta_- > 0$, then $\mathbf{v}^\top \mathbf{w}(n)$ is increasing, and $\|\mathbf{w}(n)\|$ is unbounded.*

As mentioned in previous work, it is a common phenomenon that the norm of the weight vector is infinite when data are perfectly classified. And we further verify it under the population loss and zero margin dataset.

2.2 Data distribution assumption

We need to construct a scheme with zero margin, so the data distribution should be infinite and separable. Moreover, we note that the data norm $\|\mathbf{x}\|$ does not change the corresponding label $y(\mathbf{x})$ in this case. Thus we pay our attention on $\mathcal{S}^{d-1}$. As an ideal staring point, we suppose the training data uniformly spread over $\mathcal{S}^{d-1}$, leading to our main data assumption below.

Assumption 1 *Assume $\mathbf{x} \sim \mathcal{D}$ has a spherically symmetric distribution, i.e., for any orthogonal matrix $A : A\mathbf{x} \sim \mathcal{D}$.*

[1] Here, $\mathrm{sgn}(z) = 1$ if $z \geq 0$, otherwise -1.

Spherically symmetric distributions include the standard Gaussian which is common in the literature [7, 47, 38, 48, 31]. Using this assumption, we are able to reduce the scope of optimization into the plane span$\{\mathbf{w}(0), \mathbf{v}\}$ as shown in Yehudai and Ohad [45]. Although the assumption sounds strong, we also conduct experiments in real dataset (see Section 5) to verify our discovery.

Assumption 1 also gives benign landscape property as Proposition 3 shows. Once $\mathbf{w}$ aligns with $\mathbf{v}$ (including $\mathbf{0}$), gradient methods always direct to $\mathbf{w} = r\mathbf{v}$ with $r \to +\infty$ according to Propositions 2 and 3. Therefore, we only need to consider the initial value $\mathbf{w}(0) \neq \mathbf{0}$ and $\theta(0) \neq 0$ or π. Moreover, note that c_0 does not depend on the dimension d under Assumption 1.

Proposition 3 *Under Assumption 1, we have that* $\nabla L(r\mathbf{v}) = -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \frac{|x_1|}{1+e^{r|x_1|}}\mathbf{v}, \forall\ r \in \mathbb{R}$, $\|\nabla L(\mathbf{w})\| \leq c_0$, *and* $c_0 = \mathbb{E}_{\mathbf{x}\in\mathcal{D}_{\mathbf{w},\mathbf{v}}}\|\mathbf{x}\|$ *for any* $\mathbf{w}, \mathbf{v} \in \mathbb{R}^d$.

The variation of $\theta(\mathbf{w}, \mathbf{v})$ is the most we concern, which captures population accuracy. Hence we directly consider the dynamic of the weight direction (i.e., $\overline{\mathbf{w}}$), and employ $\cos\theta(\mathbf{w}, \mathbf{v})$ in our analysis, which is also employed in previous work [29, 28, 45, 37]. In the gradient flow setting, we have

$$\frac{\partial \cos\theta(t)}{\partial t} = -\frac{1}{\|\mathbf{w}(t)\|}\left(\mathbf{v} - \left(\overline{\mathbf{w}}(t)^\top \mathbf{v}\right)\overline{\mathbf{w}}(t)\right)^\top \nabla L(\mathbf{w}(t)). \tag{2}$$

Surprisingly, we have the exact directional improvement under Assumption 1, because the symmetry eliminates the collision between data points, leading to the angle variation becoming polished.

Lemma 1 *Under Assumption 1 and if* $\mathbf{w} \neq \mathbf{0}$, *then*

$$-\left(\mathbf{v} - \left(\overline{\mathbf{w}}^\top \mathbf{v}\right)\overline{\mathbf{w}}\right)^\top \nabla L(\mathbf{w}) = \frac{c_0 \sin^2\theta(\mathbf{w}, \mathbf{v})}{\pi}.$$

Moreover, invoking from the above result, we have the following corollary:

$$-\left(I - \overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top\right)\nabla L(\mathbf{w}) = \frac{c_0}{\pi}\left(I - \overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top\right)\mathbf{v}.$$

The key proof idea of this lemma is first assuming $d = 2$ and $\overline{\mathbf{w}} = (1, 0)^\top$ using the spherical symmetry, then separating the data into two regions $\{\mathbf{x} : |v_1x_1| > |v_2x_2|\}$ and $\{\mathbf{x} : |v_1x_1| \leq |v_2x_2|\}$, while we can calculate the expectation in both regions analytically.

3 Shallow Linear Networks

In this section we begin with the case of the classical linear predictor: $\phi(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top\mathbf{x}$, and defer the detail proofs to Appendix A. Based on Eq. (2) and Lemma 1, we now turn to discuss the remaining dynamic of $\|\mathbf{w}(t)\|$ or $\|\mathbf{w}(n)\|$.

3.1 Gradient flow

From Lemma 1, we obtain that $\partial\cos\theta(t)/\partial t \geq 0$ when $\mathbf{w}(t) \neq 0$. Therefore, to derive directional convergence, we need to understand the variation of $\|\mathbf{w}(t)\|$ during optimization dynamic. We set $N(\mathbf{w}(t)) := -\mathbf{w}(t)^\top\nabla L(\mathbf{w}(t)) = \frac{1}{2}\cdot\frac{\partial\|\mathbf{w}(t)\|^2}{\partial t}$. We can obtain the variation of $\mathbf{w}(t)$ from the stable condition $N(\mathbf{w}(t)) = 0$.

Lemma 2 *Under Assumption 1, suppose* $\partial\|\mathbf{w}(t_0)\|^2/\partial t = 0$ *for some* $t_0 \geq 0$ *with* $\mathbf{w}(t_0) \neq \mathbf{0}$. *Then we have* $\partial\cos\theta(\mathbf{w}(t_0), \mathbf{v})/\partial t \geq 0$.

Therefore, based on Proposition 2 and Lemma 2, $\|\mathbf{w}(t)\|$ is first decreasing (if $\mathbf{w}(0)^\top\nabla L(\mathbf{w}(0)) < 0$) and then increasing to $+\infty$. Furthermore, note that $N(\mathbf{w}(t)) \leq 0.3$, then $\|\mathbf{w}(t)\|$ has a linear upper bound if it increases. Therefore, the directional convergence includes the following two phases.

Theorem 1 *Under Assumption 1, we obtain the following two-phase directional convergence for* $\mathbf{w}(0) \neq \mathbf{0}$ *and* $\theta(0) \neq \pi$. *If* $N(\mathbf{w}(0)) < 0$, *then there exists a finite* $T > 0$ *such that* $N(\mathbf{w}(T)) = 0$,

otherwise we set $T = 0$. With such a T, we have that

$$\cos\theta(t) \geq \begin{cases} 1 - \dfrac{2}{e^{A_1 t + B_1} + 1}, & t \leq T, \\ 1 - \dfrac{2}{e^{A_2\sqrt{t-T+C_2}+B_2} + 1}, & t > T. \end{cases}$$

Here $A_1 = \frac{2c_0}{\pi\|\mathbf{w}(0)\|}$, $B_1 = -2\ln\left|\tan\frac{\theta(0)}{2}\right|$, $A_2 = \frac{4c_0}{\sqrt{0.6}\pi}$, $B_2 = -2\ln\left|\tan\frac{\theta(T)}{2}\right| - \frac{4c_0\|\mathbf{w}(T)\|}{0.6\pi}$, $C_2 = \frac{\|\mathbf{w}(T)\|^2}{0.6}$.

The intuition behind the proof is that $\|\mathbf{w}(t)\|$ increases much slower than $\theta(t)$ decreases, showing that the objective is nearly strongly convex, which provides superpolynomial convergence.

3.2 Gradient descent

Now we turn to the gradient descent setting based on the previous results. The difficulty we encounter is that the arbitrary choice of learning rates in each step may break the directional monotonicity in Lemma 1 as Eq. (3) below reveals (See Appendix A for proofs).

$$\begin{aligned} \cos\theta\,(n{+}1) - \cos\theta\,(n) = \frac{1}{\|\mathbf{w}(n{+}1)\|}\Big[& -\eta_n \left(\mathbf{v} - (\overline{\mathbf{w}}(n)^\top \mathbf{v})\overline{\mathbf{w}}(n)\right)^\top \nabla L(\mathbf{w}(n)) \\ & - \left(\|\mathbf{w}(n{+}1)\| - \overline{\mathbf{w}}(n)^\top \mathbf{w}(n{+}1)\right)\cos\theta(n)\Big]. \end{aligned} \tag{3}$$

Fortunately, when $\mathbf{v}^\top\mathbf{w}(n) < 0$, we have $\cos\theta(n) \leq 0$ and $N(\mathbf{w}) \leq 0$, implying that the first phase directional convergence in the gradient flow case still holds.

Theorem 2 *Under Assumption 1, suppose $\theta(0) \neq \pi$ and $\mathbf{v}^\top\mathbf{w}(0) < 0$. Then we have that*

$$\cos\theta(n) \geq 1 - (1 - \cos\theta(0))\, e^{-BS_n^-}$$

until $\cos\theta(n) \geq 0$, where $S_n^- := \sum_{k=0}^{n-1} \frac{\eta_k}{\sqrt{A + \sum_{i=0}^{k}\eta_i^2}}$, $A = \frac{\|\mathbf{w}(0)\|^2}{c_0^2}$ and $B = \frac{1+\cos\theta(0)}{\pi}$.

Remark 1 *Obviously, $S_n^- < n$. And we list several choices of $\{\eta_n\}_{i=1}^\infty$:*

$$S_n^- = \begin{cases} \Theta(n), & \eta_n = \Theta(q^n), q > 1; \\ \Theta(n^{\min\{\alpha+1, 1/2\}}), & \eta_n = \Theta(n^\alpha), -1 < \alpha, \alpha \neq -1/2; \\ \Theta(\ln(n)), & \eta_n = \Theta(n^{-1}). \end{cases}$$

Hence, when $\mathbf{w}(n)$ stays in the "wrong" region that $\theta(n) > \pi/2$, larger learning rate gives faster directional convergence to the region $\{\mathbf{w} : \theta(\mathbf{w}, \mathbf{v}) \geq 0\}$. Unfortunately, when $\theta(n) \leq \pi/2$, the directional dynamic becomes unstable and heavily relies on the current learning rate. However, after simple calculation invoked from Lemma 1, we discover that

$$\|\mathbf{w}(n{+}1)\|^2 = \left(\overline{\mathbf{w}}(n)^\top \mathbf{w}(n{+}1)\right)^2 + \left(\frac{c_0\eta_n}{\pi}\right)^2 \sin^2\theta(n).$$

Combining to Eq. (3), we find out

$$\cos\theta\,(n{+}1) - \cos\theta\,(n) = \frac{1}{\|\mathbf{w}(n{+}1)\|}\left(\frac{c_0\eta_n\sin^2\theta(n)}{\pi} - \frac{c_0^2\eta_n^2\sin^2\theta(n)\cos\theta(n)/\pi^2}{\|\mathbf{w}(n{+}1)\| + \overline{\mathbf{w}}_n^\top\mathbf{w}(n{+}1)}\right).$$

Hence, we characterize a sufficient condition in the remaining training period to guarantee the directional monotonicity. Moreover, we will show that such a condition can be satisfied when the weight norm is large enough compared to the current learning rate.

Theorem 3 (A sufficient convergence condition) *Under Assumption 1, if there exists a $\delta > 0$, s.t.*

$$\|\mathbf{w}(n{+}1)\| + \overline{\mathbf{w}}(n)^\top\mathbf{w}(n{+}1) \geq \frac{(1+\delta)c_0\eta_n\cos\theta(n)}{\pi}, \forall\, n \in \mathbb{N}, \tag{4}$$

then there exist constants $A, B, C > 0$ such that

$$\cos\theta(n) \geq 1 - (1 - \cos\theta(0))\, e^{-BS_n^+}.$$

Here $S_n^+ := \sum_{k=0}^{n-1} \frac{\eta_k}{\sqrt{A + \sum_{i=0}^{k}(\eta_i^2 + C\eta_i)}}$, $A = \|\mathbf{w}(0)\|^2/c_0^2$, $B = \frac{\delta(1+\cos\theta(0))}{(1+\delta)\pi}$, and $C = 0.6/c_0^2$.

Remark 2 *Obviously, $S_n^+ < n$. And we list several choices of $\{\eta_n\}_{i=1}^{\infty}$:*

$$S_n^+ = \begin{cases} \Theta(n), & \eta_n = \Theta(q^n), q > 1; \\ \Theta(n^{(\min\{\alpha,0\}+1)/2}), & \eta_n = \Theta(n^{\alpha}), -1 < \alpha; \\ \Theta(\sqrt{\ln(n)}), & \eta_n = \Theta(n^{-1}). \end{cases}$$

Furthermore, we can also show the directional convergence with bounded learning rates as follows. Note that when $\|\mathbf{w}(n)\| \geq \eta_n c_0 + (1+\delta)c_0\eta_n/(2\pi)$, Eq. (4) can be satisfied due to

$$\|\mathbf{w}(n+1)\| + \overline{\mathbf{w}}(n)^\top \mathbf{w}(n+1) \geq 2\left(\|\mathbf{w}(n)\| - \eta_n\|\nabla L(\mathbf{w}(n))\|\right) \geq (1+\delta)c_0\eta_n/\pi.$$

Once $\eta_n \leq \eta_+$, then $\|\mathbf{w}(n)\| \geq R_1 := \eta_+ c_0 + c_0\eta_+/\pi$ is enough to derive the convergence. While from Proposition 2, $\|\mathbf{w}(n)\| \geq \mathbf{v}^\top \mathbf{w}(n)$ and the right term monotonically increases to infinity. Thus, after finite iterations, the sufficient convergence condition would be satisfied.

Theorem 4 *Under Assumption 1, for an arbitrary choice of learning rate sequence $\{\eta_n\}_{i=1}^{\infty}$ with $\eta_+ \geq \eta_n \geq \eta_- > 0$, there exists a $n_0 > 0$ such that $\mathbf{v}^\top \mathbf{w}(n_0) \geq \eta_+ c_0 + \eta_+ c_0/\pi$, and gradient descent will give superpolynomial convergence of $\cos\theta(n)$ to 1 from n_0.*

Remark 3 *Generally, a β-smooth objective function has the learning rate constraint ($\eta \leq \frac{2}{\beta}$) to guarantee the convergence. There is no such constraint because the purpose is learning the direction instead the loss. Furthermore, we need to underline that the loss still converges. Since for large enough $\|\mathbf{w}(n)\|$, the smoothness coefficient of the objective becomes certainly small and the learning rate is naturally suitable to give superpolynomial convergence.*

Comparison. Our convergence bounds provide a faster and non-asymptotic directional convergence rate $\exp\left(\mathcal{O}(-\sqrt{t})\right)$ compared to $\mathcal{O}\left(1/\log^2 t\right)$ in Soudry et al. [43]. Additionally, previous results hold for certain large t and the finite dataset with a positive margin, but we show the behavior in the whole training dynamic. We consider the possible explanation is the structured data in Assumption 1, which may be the benefit of data augmentation and preprocessing in practice. Moreover, the prior technique of proof mostly employs the decomposition as $\mathbf{w}(t) = \hat{\mathbf{w}}\log t + \boldsymbol{\rho}(t)$ with the max-margin solution $\hat{\mathbf{w}}$ and almost bounded residual term $\boldsymbol{\rho}(t)$. We can implicitly obtain an analogous decomposition, but such a way would lose the sight of variation during the early training, such as the decreasing and increasing period of $\|\mathbf{w}(t)\|$. Furthermore, for gradient descent, directional convergence in previous results is built on the loss convergence [21, 23, 33, 34, 43]. Hence they need "small" learning rates with a data-related upper bound. When data are distributed well, we derive the directional convergence directly under implicitly bounded learning rates, and then obtain the loss convergence from the directional convergence.

4 Deep Linear Networks

Invoking from the linear predictor and previous work on deep linear networks, we extend the results of gradient flow to deep linear networks and leave details in Appendix B. For an N-layer linear network $\phi(\mathbf{x}, \mathbf{w}) = W_N \ldots W_1 \mathbf{x}$ where $\mathbf{w} := (W_N, \ldots, W_1)$, the objective is

$$\min_{\mathbf{w}} L^{(N)}(W_N, \ldots, W_1) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \ln\left[1 + e^{-y(\mathbf{x})W_N \cdots W_1 \mathbf{x}}\right]. \tag{5}$$

Every such a network represents a linear mapping given as $\mathbf{w}_e = (W_N \cdots W_1)^\top \in \mathbb{R}^d$:

$$L^{(N)}(W_1, \ldots, W_N) = L^{(1)}(\mathbf{w}_e) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \ln\left(1 + e^{-y(\mathbf{x})\mathbf{w}_e^\top \mathbf{x}}\right).$$

A key tool for analyzing the induced flow for $\mathbf{w}_e$ is established in Claim 2 of Arora et al. [3]. If the initial balancedness conditions

$$W_{j+1}(0)^\top W_{j+1}(0) = W_j(0) W_j(0)^\top, j = 1, \ldots, N-1 \tag{6}$$

hold, then we have the induced gradient flow with $\mathbf{w}_e(t)$:

$$\frac{\partial \mathbf{w}_e(t)}{\partial t} = -\|\mathbf{w}_e(t)\|^{2-\frac{2}{N}} \left(\frac{dL^{(1)}(\mathbf{w}_e(t))}{d\mathbf{w}} + (N-1)\overline{\mathbf{w}}_e(t)\overline{\mathbf{w}}_e(t)^\top \frac{dL^{(1)}(\mathbf{w}_e(t))}{d\mathbf{w}}\right). \tag{7}$$

Similarly, we can build up the monotonic directional improvement in the following lemma.

Lemma 3 *Under Assumption 1 and the initial balancedness condition Eq. (6), if* $\mathbf{w}_e(t) \neq \mathbf{0}$*, then*

$$\frac{\partial \cos\theta(\mathbf{w}_e(t), \mathbf{v})}{\partial t} = \frac{c_0 \sin^2\theta(\mathbf{w}_e(t), \mathbf{v})}{\pi} \cdot \|\mathbf{w}_e(t)\|^{1-\frac{2}{N}} . \tag{8}$$

The main difference from the shallow linear network is that the dependence of weight norm $\|\mathbf{w}_e(t)\|$ is reversed. Larger $\|\mathbf{w}_e(t)\|$ gives faster convergence for the deep linear networks when $N \geq 3$, while no influence for $N = 2$, and for $N = 1$ it is clearly opposite.

Remark 4 *Lemma 3 shows three possible facts in practice. Note that* $\theta(\mathbf{w}_e(t), \mathbf{v})$ *is proportional to accuracy in our scheme, and* c_0 *is the expectation norm of data. Therefore, enlarging a) the data norm (*c_0*), or b) the induced weight norm (*$\|\mathbf{w}_e(t)\|$*), or c) the depth (*N*) could accelerate the training. The intuitive of a) is that larger data make the gradient penalizes more on wrong data, showing more radical variation as increasing learning rates. Meanwhile, b) can also be viewed as increasing learning rates because the gradient of each layer weight is the multiply of the other layer weights. Finally, c) has already rigorously examined in Arora et al. [3].*

Thanks to the similar expression of the induced weight gradient, with the technique introduced in the linear predictor, we can still construct the variation of $\|\mathbf{w}_e(t)\|$ during optimization.

Lemma 4 *Under Assumption 1 and the initial balancedness condition Eq. (6) with* $\mathbf{w}_e(0) \neq \mathbf{0}$*, we have the following two properties:*

- *Suppose* $N > 2$*. Then we have*

$$\left(\|\mathbf{w}_e(0)\|^{\frac{2}{N}} + 0.6t\right)^{\frac{N}{2}} \geq \|\mathbf{w}_e(t)\| \geq \left(\|\mathbf{w}_e(0)\|^{\frac{2}{N}-1} + (N-2)c_0 t\right)^{-\frac{N}{N-2}} > 0. \tag{9}$$

- *Suppose* $\partial\|\mathbf{w}_e(t_0)\|^2/\partial t = 0$ *for some* $t_0 \geq 0$*. Then we have* $\partial \cos\theta(\mathbf{w}_e(t_0), \mathbf{v})/\partial t \geq 0$*.*

Using Lemma 4, we can still obtain descent-ascent transition of the weight norm. The only potential difficulty is that $\mathbf{w}_e(t)$ may converge to the potential stationary point at the origin (at which the angle is not well-defined). While following Eq. (9) in Lemma 4, $\|\mathbf{w}_e(t)\|$ only could converge to zero in infinite time. Based on the first period directional convergence in Theorem 5 below, $\cos\theta(t)$ would always increase and becomes positive after finite time. Thus we have $N(\mathbf{w}_e(t)) \geq 0$ for small enough $\|\mathbf{w}_e(t)\|$, leading to the increasing of $\|\mathbf{w}_e(t)\|$. Finally, $\|\mathbf{w}_e(t)\|$ continues increasing based on the second point of Lemma 4. Therefore, $\|\mathbf{w}_e(t)\|$ could not converge to zero, leading to the two-phase directional convergence as well.

Theorem 5 *Under Assumption 1 and the initial balancedness condition Eq. (6), if* $N > 2$*,* $\mathbf{w}_e(0) \neq \mathbf{0}$ *and* $\theta(0) \neq \pi$*, we obtain two-phase convergence as follows. If* $\partial\|\mathbf{w}_e(0)\|^2/\partial t < 0$*, then there exists a finite* $T > 0$ *such that* $\partial\|\mathbf{w}_e(T)\|^2/\partial t = 0$*, otherwise, we set* $T = 0$*. With such a* T*, it holds that*

$$\cos\theta(t) \geq \begin{cases} 1 - \dfrac{2}{C_1(A_1 t/B_1 + 1)^{\alpha} + 1}, & t \leq T, \\ 1 - \dfrac{2}{e^{A_2(t-T)+B_2} + 1}, & t \geq T. \end{cases}$$

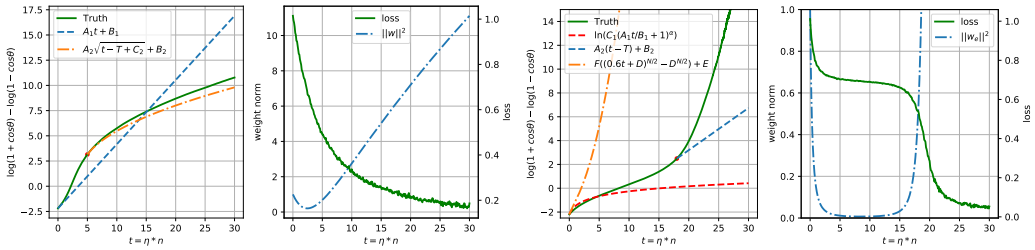
Here $A_1 = (N-2)c_0$*,* $B_1 = \|\mathbf{w}_e(0)\|^{\frac{2}{N}-1}$*,* $C_1 = \frac{1+\cos\theta(0)}{1-\cos\theta(0)}$*, and* $A_2 = 2c_0\|\mathbf{w}_e(T)\|^{2-\frac{2}{N}}/\pi$*,* $B_2 = -2\ln\left|\tan\frac{\theta(T)}{2}\right|$*,* $\alpha = 2c_0/\pi$*.*

In addition, we have the upper bound that

$$\cos\theta(t) \leq 1 - \frac{2}{e^{F[(0.6t+D)^{N/2} - D^{N/2}]+E} + 1},$$

where $D = \|\mathbf{w}_e(0)\|^{\frac{2}{N}}$*,* $E = -2\ln\left|\frac{\tan\theta(0)}{2}\right|$*,* $F = \frac{4c_0}{0.6N\pi}$*.*

As N increases, $A_1 = (N-2)c_0$ also increases. Then $\cos\theta(t)$ converges faster, which is consistent with the implicit acceleration of deep networks shown in Arora et al. [3]. As for the initialization $\theta(0) = \pi$, we have that $\theta(t) = \pi,\ \forall t \geq 0$, and $\mathbf{w}_e(t) \to \mathbf{0}$ but never hits the origin.



(a) Linear predictor

(b) Four-layer deep linear network

Figure 1: Simulation of (deep) linear network with $\mathbf{x} \sim \mathcal{U}(\mathcal{S}^1)$, $\mathbf{v} = (0,1)^\top$. (a): Linear predictor. We show our lower bounds in Theorem 1 at $n = 0$ and $n = 5000$. (b): Four-layer deep linear network. We also show our lower and upper bounds in Theorem 5 at $n = 0$ and $n = 18000$. In each experiment, we plot angle variation, loss and weight norm in sequence.

Remark 5 *Since we need to cover the worse case during optimization, our bound may seem loose in two possible cases. First, when $t \le T$, we use the lower bound in Eq. (9) to capture the decreasing period of $\|\mathbf{w}_e(t)\|$. Actually, $\|\mathbf{w}_e(t)\|$ may descend to certain large norm then ascend. Second, when $t \ge T$, note that $\theta(t) \to 1$ except $\mathbf{w}_e(0) = k\mathbf{v}$ for $k \le 0$. We can also guarantee $\|\mathbf{w}_e(t)\| \to \infty$ because $\mathbf{v}^\top \mathbf{w}_e(t)$ is increasing after some time (but not always). The convergence could be faster when $\|\mathbf{w}_e(t)\|$ increases much, but we only treat such a scheme lower bounded by $\|\mathbf{w}_e(T)\|$.*

Comparison. Our concise convergence results for deep linear networks neither require small training loss [32, 24, 22], nor presumed convergence of loss and weight direction [17, 33], nor overparameterization [10, 11, 19]. However, we need the initial balancedness condition [3], while Shamir [41], Hu et al. [19] showed the necessity of good initialization. Moreover, we obtain explicit directional convergence particularly for the early training dynamic while previous works [32, 24, 46] mainly provide asymptotic or late training convergence, though our results hold under the benign training data and limited network structure compared to Lyu and Li [32], Yun et al. [46].

5 Experiments

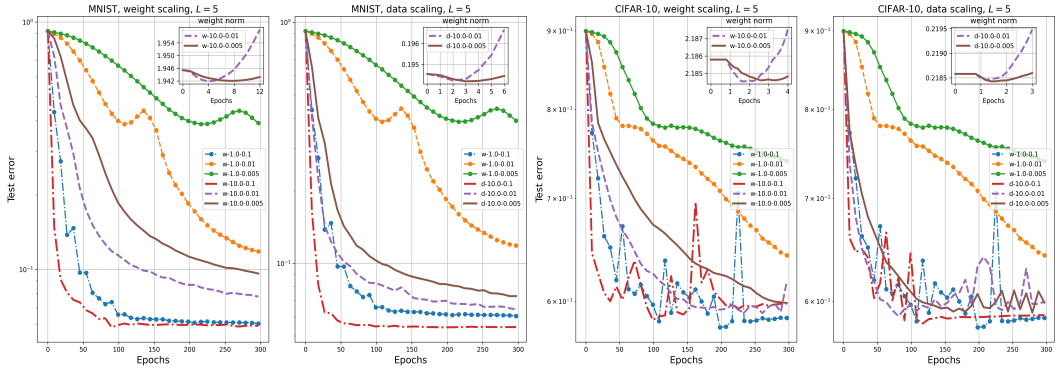
In this section we conduct experiments to verify our theoretical analyses. We also study more general settings under real datasets with our theoretical insights.

Linear networks. We first consider training a linear classifier and a 4-layer linear network under the logit loss as our theorems show. We construct simple dataset with $\mathbf{x} \sim \mathcal{U}(\mathcal{S}^1)$ and $y = \mathbf{v}^\top \mathbf{x}$ with $\mathbf{v} = (0,1)^\top$. We use common stochastic gradient descent (SGD) with batch size 1000 and small learning rate 10^{-3}. Moreover, we choose a initialization $\mathbf{w}(0) = \mathbf{w}_e(0) = (0.6, -0.8)^\top$. In the deep linear network, we set $W_N(0) = \boldsymbol{u}_N^\top$, $W_i(0) = \boldsymbol{u}_{i+1}\boldsymbol{u}_i^\top, i = 1, \ldots, N-1$, with $\|\boldsymbol{u}_i\| = 1$ and $\boldsymbol{u}_1 = \mathbf{w}_e(0)$ to satisfy the balancedness conditions Eq. (6). The results are shown in Figure 1.

Figure 1a shows the optimization period for linear classifiers. We also plot the convergence bounds obtained in Theorems 1. Although we do not give convergence for SGD, our bounds in gradient flow still roughly matches the directional convergence in practice, and the weight norm $\|\mathbf{w}(n)\|$ indeed goes through clearly decreasing and increasing period.

Figure 1b depicts the dynamic for the deep linear networks. We also plot three convergence bounds in Theorems 5, which roughly match the actual behavior. The weight norm $\|\mathbf{w}(n)\|$ still goes through the decreasing and increasing period, and we could observe a distinct stuck period when $5 \le \eta \cdot n \le 10$ as the lower bound shown in Lemma 4. Moreover, the loss also decreases slowly when $\|\mathbf{w}(n)\|$ is certainly small, but $\cos\theta(t)$ still has a considerable growth. Thus previous loss-based analysis may neglect potential variation of the accuracy we most concern. Furthermore, we can see rigorous understanding of $\|\mathbf{w}(n)\|$ may give more precise rates in the late training period ($t \ge 20$).

Non-linear networks. As we discuss earlier in Remark 4, increasing the data norm, or the weight norm, or the depth of network could accelerate training process from Eq. (8). Now we briefly verify the first two perspectives through L-layer fully-connected networks with ReLU activation trained



(a) MNIST, $L = 5$. (b) CIFAR-10, $L = 5$.

Figure 2: Simulation of deep fully-connected ReLU networks. The dataset used, the accelerating strategy (weight scaling or data scaling) and depth are shown in the title of each subgraph. We choose different scaling factors α and learning rates η by showing the legends as 'm-α-η', that m=w is weight scaling and m=d is data scaling. For example: 'w-10.0-0.01' is using weight scaling with factor 10.0 and learning rate 0.01. And 'w-1.0-$\cdots$' is the default initialization. Some of the variations of induced weight norm $\|\mathbf{w}_e\|$ are shown in the sub-picture in each subgraph.

on the well-known MNIST and CIFAR-10 datasets [27, 26]. We set the layer width as 100 across all hidden-layers with no bias and regularization terms under cross entropy loss. We also randomly choose a subset of the original dataset with 5000 training samples and run SGD using batch size 1000 with constant learning rates selected from $0.1, 0.01, 0.005$. We employ default PyTorch initialization for each layer [36]. After the original initialization, we multiply 1) each input training data with a scaling factor α, or 2) each layer weight $W_i, i \in \{1, \ldots, L\}$ with $\alpha^{1/L}$ to produce different norm of training data or induced weight $w_e = (W_L \cdots W_1)^\top$. The results for 5-layer DNN with $\alpha \in \{1, 10\}$ are shown in Figure 2, and remaining experimental results are shown in Appendix C.

In Figure 2, we observe that the dynamics of test error (one minus test accuracy) employed with weight scaling or data scaling are faster than default case in the early training phase, while the superiority is weak when large learning rates employed. This however does not serve as evidence in favor of acceleration by weight scaling as we did not set learning rates optimally per model, while weight scaling already affects the learning rates. As for data scaling, the acceleration is convincing because the learning rates are the same. Moreover, we discover the descending and ascending behavior in the sub-picture of each figure only for small learning rates and scaled data or weight. Generally, we could not see such behaviors through frequently-used learning rates (larger than 0.01 for GD methods) and default initialization, in which the induced norm continues increasing. Furthermore, we perceive the linear convergence in the beginning, but soon some of them bump around their best performance when large learning rates used, or have different convergent phases under small learning rates. Therefore, we think the datasets do not satisfy our data assumption generally, but the convincing early training behavior may be seen as an interesting problem.

6 Conclusion

In this work, we have studied the behavior of gradient flow and gradient descent on zero-margin separable data under population loss in binary linear classification tasks. We have proved exact directional convergence for (deep) linear networks in the whole training dynamic on spherically symmetric data. Moreover, we have characterized the descent and ascent of induced weight norm theoretically, and also verified such phenomena in the numerical experiments and real datasets. Although our data assumption is idealized, we discover (linear) directional improvements in the early training phase for non-linear activations and real datasets, which we would figure out theoretically in future work. Actually, we have derived the directional convergence for a two-layer ReLU network with only two hidden neurons in Appendix D. We hope that our specific view of directional convergence would bring better understanding of the optimization dynamics of gradient methods on neural networks in classification tasks.

References

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[2] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2018.

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

[4] 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*, pages 322–332. PMLR, 2019.

[5] Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E Schapire. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[6] Peter Bartlett, Dave Helmbold, and Philip Long. Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks. In *International conference on machine learning*, pages 521–530. PMLR, 2018.

[7] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *International conference on machine learning*, pages 605–614. PMLR, 2017.

[8] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846, 2019.

[9] Yuan Cao and Quanquan Gu. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3349–3356, 2020.

[10] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[11] Simon Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. In *International Conference on Machine Learning*, pages 1655–1664. PMLR, 2019.

[12] 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*, pages 1675–1685. PMLR, 2019.

[13] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

[14] Armin Eftekhari. Training linear neural networks: Non-local convergence and complexity results. In *International Conference on Machine Learning*, pages 2836–2847. PMLR, 2020.

[15] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[16] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018.

[17] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pages 9461–9471, 2018.

[18] Jeff Z. HaoChen, Colin Wei, Jason D. Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance, 2021. URL `https://openreview.net/forum?id=crAi7c41xTh`.

[19] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *International Conference on Learning Representations*, 2019.

[20] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[21] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pages 1772–1798. PMLR, 2019.

[22] Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[23] Ziwei Ji and Matus Telgarsky. Characterizing the implicit bias via a primal-dual analysis. In *Algorithmic Learning Theory*, pages 772–804. PMLR, 2021.

[24] Ziwei Ji and Matus Jan Telgarsky. Gradient descent aligns the layers of deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[25] Ziwei Ji, Miroslav Dudík, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *Conference on Learning Theory*, pages 2109–2136. PMLR, 2020.

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Chris Junchi Li, Mendgi Wang, Han Liu, and Tong Zhang. Diffusion approximations for online principal component estimation and global convergence. *Advances in Neural Information Processing Systems*, page 646, 2017.

[29] Chris Junchi Li, Mengdi Wang, Han Liu, and Tong Zhang. Near-optimal stochastic approximation for online principal component estimation. *Mathematical Programming*, 167(1):75–97, 2018.

[30] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8157–8166, 2018.

[31] Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang. Learning over-parametrized two-layer neural networks beyond ntk. In *Conference on Learning Theory*, pages 2613–2682. PMLR, 2020.

[32] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2019.

[33] Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Pedro Henrique Pamplona Savarese, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3420–3428. PMLR, 2019.

[34] Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3051–3059. PMLR, 2019.

[35] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[37] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR, 2019.

[38] Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks. In *International Conference on Machine Learning*, pages 4433–4441. PMLR, 2018.

[39] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

[40] Shai Shalev-Shwartz and Yoram Singer. On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. *Machine learning*, 80(2): 141–163, 2010.

[41] Ohad Shamir. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In *Conference on Learning Theory*, pages 2691–2713. PMLR, 2019.

[42] Ohad Shamir. Gradient methods never overfit on separable data. *Journal of Machine Learning Research*, 22(85):1–20, 2021.

[43] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[44] 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*, pages 3635–3673. PMLR, 2020.

[45] Gilad Yehudai and Shamir Ohad. Learning a single neuron with gradient methods. In *Conference on Learning Theory*, pages 3756–3786. PMLR, 2020.

[46] Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=ZsZM-4iMQkH`.

[47] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pages 4140–4149. PMLR, 2017.

[48] Mo Zhou, Rong Ge, and Chi Jin. A local convergence theory for mildly over-parameterized two-layer neural network. *arXiv preprint arXiv:2102.02410*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] The limitations mainly come from the strong assumption of spherically symmetric distribution in Assumption 1, but numerous previous works employ the standard Gaussian data [7, 47, 38, 48, 31], which is a subcase of ours. Moreover, we also try non-linear network with theories on two neurons (Appendix D) and experiments in Section 5. We consider our results still match early training results and descent-ascent induced weight norm variation.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] The purpose of this work is to provide a better understanding of neural network optimization.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix for detail.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code are proprietary, but we will dispaly the URL after we clear up.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 for description.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] But we show other experimental results in Appendix C.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We do not use large datasets and the experiments is certainly simple and easy to follow.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] These datasets are common in works.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] These datasets are common in works.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] These datasets are common in works.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]