Generalization Error Bounds of Gradient Descent for Learning Over-parameterized Deep ReLU Networks

Yuan Cao and Quanquan Gu
Department of Computer Science
University of California, Los Angeles
{yuancao,qgu}@cs.ucla.edu

Abstract

Empirical studies show that gradient-based methods can learn deep neural networks (DNNs) with very good generalization performance in the over-parameterization regime, where DNNs can easily fit a random labeling of the training data. Very recently, a line of work explains in theory that with over-parameterization and proper random initialization, gradient-based methods can find the global minima of the training loss for DNNs. However, existing generalization error bounds are unable to explain the good generalization performance of over-parameterized DNNs. The major limitation of most existing generalization bounds is that they are based on uniform convergence and are independent of the training algorithm. In this work, we derive an algorithm-dependent generalization error bound for deep ReLU networks, and show that under certain assumptions on the data distribution, gradient descent (GD) with proper random initialization is able to train a sufficiently over-parameterized DNN to achieve arbitrarily small generalization error. Our work sheds light on explaining the good generalization performance of over-parameterized deep neural networks.

1 Introduction

Deep learning achieves great successes in almost all real-world applications ranging from image processing (Krizhevsky, Sutskever, and Hinton 2012), speech recognition (Hinton et al. 2012) to Go games (Silver et al. 2016). Understanding and explaining the success of deep learning has thus become a central problem for theorists. One of the mysteries is that the neural networks used in practice are often heavily over-parameterized such that they can even fit random labels to the input data (Zhang et al. 2017), while they can still achieve very small generalization error (i.e., test error) when trained with real labels.

There are multiple recent attempts towards answering the above question and demystifying the success of deep learning. (Soudry and Carmon 2016; Safran and Shamir 2016; Arora, Cohen, and Hazan 2018; Haeffele and Vidal 2015; Nguyen and Hein 2017) showed that over-parameterization can lead to better optimization landscape. (Li and Liang 2018; Du et al. 2019b) proved that with proper random initialization, gradient descent (GD) and/or stochastic gradient descent (SGD) provably find the global minimum for training over-parameterized one-hidden-layer ReLU networks. (Arora et al. 2018a) analyzed the convergence of GD to global optimum for training a deep linear neural network under a set of assumptions on the network width and initialization. (Du et al. 2019a; Allen-Zhu, Li, and Song 2019; Zou et al. 2019) studied the convergence of gradient-based method for training over-parameterized deep nonlinear neural networks. Specifically, (Du et al. 2019a) proved that gradient descent can converge to the global minima for over-parameterized deep neural networks with smooth activation functions. (Allen-Zhu, Li, and Song 2019; Zou et al. 2019) independently proved the global convergence results of GD/SGD for deep neural networks with ReLU activation functions in the over-parameterization regime. However, in such an over-parametrized regime, the training loss function of deep neural networks may have potentially infinitely many global minima, but not all of them can generalize well. Hence, convergence to the global minimum of the training loss is not sufficient to explain the good generalization performance of GD/SGD.

There are only a few studies on the generalization theory for learning neural networks in the over-parameterization regime. (Brutzkus et al. 2018) showed that SGD learns over-parameterized networks that provably generalize on linearly separable data. (Song, Montanari, and Nguyen 2018) showed that when training two-layer networks in a suitable scaling limit, the SGD dynamic is captured by a certain non-linear partial differential equation with nearly ideal generalization error. (Li and Liang 2018) relaxed the linear separable data assumption and proved that SGD learns an over-parameterized network with a small generalization error when the data comes from mixtures of well-separated distributions. (Allen-Zhu, Li, and Liang 2019) proved that under over-parameterization, SGD or its variants can learn some notable hypothesis classes, including two and three-layer neural networks with fewer parameters. (Arora et al. 2019) provided a generalization bound of GD for two-layer ReLU networks based on a fine-grained analysis on how much the network parameters can move during GD. Nev-

ertheless, all these results are limited to two or three layer neural networks, and cannot explain the good generalization performance of gradient-based methods for *deep* neural networks. For deep neural networks, existing generalization error bounds (Neyshabur, Tomioka, and Srebro 2015; Bartlett, Foster, and Telgarsky 2017; Neyshabur et al. 2018a; Golowich, Rakhlin, and Shamir 2018; Dziugaite and Roy 2017; Arora et al. 2018b; Li et al. 2018; Neyshabur et al. 2018b; Wei et al. 2019) are mostly based on uniform convergence and independent of the training algorithms. (Daniely 2017) established a generalization bound for over-parameterized neural networks trained with one-pass SGD. However, they considered a setting where the training of hidden layers are neglectable and only the output layer training is effective.

In this paper, we aim to answer the following question:

Why gradient descent can learn an over-parameterized deep neural network that generalizes well?

Specifically, we consider learning deep fully connected ReLU networks with cross-entropy loss using over-parameterization and gradient descent.

1.1 Our Main Results and Contributions

The following theorem gives an informal version of our main results.

Theorem 1.1 (Informal version of Corollaries 3.2,3.3)**.** *Under certain data distribution assumptions, for any $\epsilon > 0$, if the number of nodes per each hidden layer is set to $\widetilde{\Omega}(\epsilon^{-14})$ and the sample size $n = \widetilde{\Omega}(\epsilon^{-4})$, then with high probability, gradient descent with properly chosen step size and random initialization method learns a deep ReLU network and achieves a population classification error at most ϵ.*

Here in Theorem 1.1 we use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to hide some logarithmic terms in standard Big-O and Big-Omega notations. The result of Theorem 1.1 holds for ReLU networks with arbitrary constant number of layers, as long as the data distribution satisfies certain separation condition, which will be discussed in Section 3.2.

Our contributions. Our main contributions are as follows:

- We provide a generalization error bound specifically suitable for wide neural networks of arbitrary depth. The bound enjoys better dependency in terms of the network width compared with existing generalization error bounds for deep neural networks (Neyshabur, Tomioka, and Srebro 2015; Bartlett, Foster, and Telgarsky 2017; Neyshabur et al. 2018a; Golowich, Rakhlin, and Shamir 2018; Arora et al. 2018b; Li et al. 2018; Wei et al. 2019). Moreover, we also provide an optimization result on the convergence of gradient descent for over-parameterized neural networks. Combining these two results together gives an algorithm dependent bound of expected error that is independent of the network width.
- We investigate two types of data distribution assumptions, and show that under each of them, gradient descent can train an over-parameterized neural network to achieve ϵ expected error provided $\widetilde{O}(\epsilon^{-4})$ training examples. The data distribution assumptions we consider in this paper are standard and have been studied in recent literature. This demonstrates that our analysis can give meaningful generalization bounds even for very wide neural networks, and can provide insights on the practical success of over-parameterized neural networks.

1.2 Notation

Throughout this paper, scalars, vectors and matrices are denoted by lower case, lower case bold face, and upper case bold face letters respectively. For a positive integer n, we denote $[n] = \{1, \ldots, n\}$. For a vector $\mathbf{x} = (x_1, \ldots, x_d)^\top$, we denote by $\|\mathbf{x}\|_p = \big(\sum_{i=1}^d |x_i|^p\big)^{1/p}$, $\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,d} |x_i|$, and $\|\mathbf{x}\|_0 = |\{x_i : x_i \neq 0, i = 1, \ldots, d\}|$ the ℓ_p, ℓ_∞ and ℓ_0 norms of $\mathbf{x}$ respectively. We use $\mathrm{Diag}(\mathbf{x})$ to denote a square diagonal matrix with the entries of $\mathbf{x}$ on the main diagonal. For a matrix $\mathbf{A} = (A_{ij}) \in \mathbb{R}^{m\times n}$, we use $\|\mathbf{A}\|_2$ and $\|\mathbf{A}\|_F$ to denote the spectral norm (maximum singular value) and Frobenius norm of $\mathbf{A}$ respectively. We also denote by $\|\mathbf{A}\|_0$ the number of nonzero entries of $\mathbf{A}$. We denote by $S^{d-1} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1\}$ the unit sphere in $\mathbb{R}^d$. For a function $f : \mathbb{R}^d \to \mathbb{R}$, we denote by $\|f(\cdot)\|_\infty = \inf\{C \geqslant 0 : |f(\mathbf{x})| \leqslant C \text{ for almost every } \mathbf{x}\}$ the essential supreme of f.

We use the following standard asymptotic notations. For two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n = O(b_n)$ if $a_n \leqslant C_1 b_n$ for some absolute constant $C_1 > 0$, and $a_n = \Omega(b_n)$ if $a_n \geqslant C_2 b_n$ for some absolute constant $C_2 > 0$. In addition, we use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to hide some logarithmic terms in Big-O and Big-Omega notations.

2 Problem Setup and Training Algorithm

In this paper, for the sake of simplicity, we study the binary classification problem on some unknown but fixed data distribution $\mathcal{D}$ over $\mathbb{R}^d \times \{+1, -1\}$. An example $(\mathbf{x}, y)$ drawn from $\mathcal{D}$ consists of the input $\mathbf{x} \in \mathbb{R}^d$ and output label $y \in \{+1, -1\}$. We denote by $\mathcal{D}_\mathbf{x}$ the marginal distribution of $\mathbf{x}$. Given an input $\mathbf{x}$, we consider predicting its corresponding label y using a deep neural network with the ReLU activation function $\sigma(z) := \max\{0, z\}$. We consider L-hidden-layer neural networks with m_l hidden nodes on the l-th layer for $l = 1, \ldots, L$. The neural network function (mapping) is defined as follows

$$f_\mathbf{W}(\mathbf{x}) = \mathbf{v}^\top \sigma(\mathbf{W}_L^\top \sigma(\mathbf{W}_{L-1}^\top \cdots \sigma(\mathbf{W}_1^\top \mathbf{x}) \cdots)),$$

where $\sigma(\cdot)$ denotes the entry-wise ReLU activation function (with a slight abuse of notation), $\mathbf{W}_l = (\mathbf{w}_{l,1}, \ldots, \mathbf{w}_{l,m_l}) \in \mathbb{R}^{m_{l-1}\times m_l}$, $l = 1, \ldots, L$ are the weight matrices, and $\mathbf{v} \in (\mathbf{1}^\top, -\mathbf{1}^\top)^\top \in \{-1, +1\}^{m_L}$ is the fixed output layer weight vector with half 1 and half -1 entries. In particular, set $m_0 = d$. We denote by $\mathbf{W} = \{\mathbf{W}_l\}_{l=1}^L$ the collection of matrices $\mathbf{W}_1, \ldots, \mathbf{W}_L$.

Given n training examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ drawn independently from $\mathcal{D}$, the training of the neural network can be formulated as an empirical risk minimization (ERM) problem as follows:

$$\min_\mathbf{W} L_S(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^n \ell[y_i \cdot f_\mathbf{W}(\boldsymbol{x}_i)], \tag{2.1}$$

where $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ is the training sample set, and $\ell(z)$ is the loss function. In this paper, we focus on cross-entropy loss function, which is in the form of $\ell(z) = \log[1 + \exp(-z)]$. Our result can be extended to other loss functions such as square loss and hinge loss as well.

2.1 Gradient Descent with Gaussian Initialization

Here we introduce the details of the algorithm we use to solve the empirical risk minimization problem (2.1). The entire training algorithm is summarized in Algorithm 1.

Algorithm 1 Gradient descent for DNNs starting at Gaussian initialization

Require: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, number of iterations K, step size η.

Generate each entries of $\mathbf{W}_l^{(0)}$ independently from $N(0, 2/m_l)$, $l \in [L]$.

for $k = 0, 1, 2, \ldots, K-1$ **do**

$\mathbf{W}_l^{(k)} = \mathbf{W}_l^{(k-1)} - \eta \nabla_{\mathbf{W}_l} L_S(\mathbf{W}_l^{(k-1)})$, $l \in [L]$.

end for

$k^* = \mathrm{argmin}_{k \in \{0,\ldots,K-1\}} -\frac{1}{n} \sum_{i=1}^n \ell'\big(y_i \cdot f_{\mathbf{W}}^{(k)}(\boldsymbol{x}_i)\big)$.

Ensure: $\mathbf{W}^{(0)}, \ldots, \mathbf{W}^{(K)}$

In detail, Algorithm 1 consists of two stages: random initialization and gradient descent (GD). In the random initialization stage, we initialize $\mathbf{W}^{(0)} = \{\mathbf{W}_l^{(0)}\}_{l=1}^L$ via Gaussian initialization for all $l \in [L]$, where each entries of $\mathbf{W}_l^{(0)}$ are generated independently from $N(0, 2/m_l)$. Note that the initialization scheme of $\mathbf{W}^{(0)}$ is essentially the initialization proposed in (He et al. 2015). In the gradient descent stage, we do gradient descent starting from $\mathbf{W}^{(0)}$, where $\eta > 0$ is the step size, and the superscript (k) is the iteration index of GD. One can also use stochastic gradient descent (SGD) to solve (2.1), and our theory can be extended to SGD as well. Due to space limit, we only consider GD in this paper.

3 Main Theory

In this section we present our main result. We first introduce several assumptions.

Assumption 3.1. *The input data are normalized:* $\mathrm{supp}(\mathcal{D}_x) \subseteq S^{d-1}$.

Assumption 3.1 is widely made in most existing work on over-parameterized neural networks (Li and Liang 2018; Allen-Zhu, Li, and Song 2019; Du et al. 2018; 2019b; Zou et al. 2019). This assumption can be relaxed to the case that $c_1 \leqslant \|\mathbf{x}\|_2 \leqslant c_2$ for all $\mathbf{x} \in \mathrm{supp}(\mathcal{D}_x)$, where $c_2 > c_1 > 0$ are absolute constants. Such relaxation will not affect our final generalization results.

Assumption 3.2. *We have* $M/m = O(1)$, *where* $M = \max\{m_1, \ldots, m_L\}$, $m = \min\{m_1, \ldots, m_L\}$.

Assumption 3.2 essentially says that the width of each layer in the deep neural network is in the same order, and the neural work architecture is balanced. Throughout this paper, we always assume Assumptions 3.1 and 3.2 hold. We therefore omit them in our theorem statements.

For the ease of exposition we introduce the following definitions.

Definition 3.1. *For the collection of random parameters* $\mathbf{W}^{(0)} = \{\mathbf{W}_l^{(0)}\}_{l=1}^L$ *generated in Algorithm 1, we call*

$$\mathcal{W}_\tau := \big\{\mathbf{W} = \{\mathbf{W}_l\}_{l=1}^L : \|\mathbf{W}_l - \mathbf{W}_l^{(0)}\|_F \leqslant \tau,\ l \in [L]\big\}$$

the τ*-neighborhood of* $\mathbf{W}^{(0)}$.

The definition of $\mathcal{W}_\tau$ is motivated by the observation that in a small neighborhood of initialization, deep ReLU networks satisfy good scaling and landscape properties. It also provides a small subset of the entire hypothesis space and enables a sharper capacity bound based on Rademacher complexity for the generalization gap between empirical and generalization errors.

Definition 3.2. *For a collection of parameter matrices* $\mathbf{W} = \{\mathbf{W}_l\}_{l=1}^L$, *we define its empirical surrogate error* $\mathcal{E}_S(\mathbf{W})$ *and population surrogate error* $\mathcal{E}_{\mathcal{D}}(\mathbf{W})$ *as follows:*

$$\mathcal{E}_S(\mathbf{W}) := -\frac{1}{n} \sum_{i=1}^n \ell'\big[y_i \cdot f_{\mathbf{W}}(\boldsymbol{x}_i)\big],$$

$$\mathcal{E}_{\mathcal{D}}(\mathbf{W}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\big\{-\ell'\big[y \cdot f_{\mathbf{W}}(\mathbf{x})\big]\big\}.$$

The intuition behind the definition of surrogate error is that, for cross-entropy loss we have $-\ell'(z) = 1/[1 + \exp(z)]$, which can be seen as a smooth version of the indicator function $\mathbb{1}\{z < 0\}$, and therefore $-\ell'[y \cdot f_{\mathbf{W}}(\mathbf{x})]$ is related to the classification error of the neural network. Surrogate error plays a pivotal role in our generalization analysis: on the one hand, it is closely related to the derivative of the empirical loss function. On the other hand, by $-2\ell'(z) \geqslant \mathbb{1}\{z < 0\}$, it also provides an upper bound on the classification error. It is worth noting that the surrogate error is comparable with the ramp loss studied in margin-based generalization error bounds (Neyshabur, Tomioka, and Srebro 2015; Bartlett, Foster, and Telgarsky 2017; Neyshabur et al. 2018a; Golowich, Rakhlin, and Shamir 2018; Arora et al. 2018b; Li et al. 2018) in the sense that it is Lipschitz continuous in $\mathbf{W}$, which ensures that $\mathcal{E}_S(\mathbf{W})$ concentrates on $\mathcal{E}_{\mathcal{D}}(\mathbf{W})$ uniformly over the parameter space $\mathcal{W}_\tau$.

3.1 Generalization and Optimization of Over-parameterized Neural Networks

In this section, we provide (i) a generalization bound for neural networks with parameters in a neighborhood of random initialization, (ii) a convergence guarantee of gradient descent for training over-parameteried neural networks. Combining these two results gives a bound on the expected error of neural networks trained by gradient descent.

Theorem 3.1. *For any* $\delta > 0$, *there exist absolute constants* $\overline{C}, \overline{C}', \underline{C}$ *such that, if*

$$m \geqslant \overline{C} \max\{L^2 \log(mn/\delta), L^{-8/3}\tau^{-4/3} \log[m/(\tau\delta)]\},$$

$$\tau \leqslant \underline{C} L^{-6}[\log(m)]^{-3/2},$$

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

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big[y \cdot f_{\mathbf{W}}(\mathbf{x}) < 0\big] \leqslant 2 \cdot \mathcal{E}_S(\mathbf{W}) + \overline{C}'\big[L\tau \cdot \sqrt{m/n} + L^4\sqrt{m\log(m)}\tau^{4/3}\big]$$

for all $\mathbf{W} \in \mathcal{W}_\tau$.

Remark 3.1. *For neural networks initialized with He initialization(He et al. 2015), the generalization bound given by Theorem 3.1 has a better dependency in network width m compared with existing uniform convergence based generalization error bounds (Neyshabur, Tomioka, and Srebro 2015; Bartlett, Foster, and Telgarsky 2017; Neyshabur et al. 2018a; Golowich, Rakhlin, and Shamir 2018; Arora et al. 2018b; Li et al. 2018; Wei et al. 2019). For instance, $\mathbf{W} \in \mathcal{W}_\tau$ implies $\|\mathbf{W}_l^\top - \mathbf{W}_l^{(0)\top}\|_{2,1} \leqslant \sqrt{m}\tau$ and $\|\mathbf{W}_l\|_2 = \widetilde{O}(1)$. Plugging these bounds into the generalization bound given by (Bartlett, Foster, and Telgarsky 2017)*

$$\widetilde{O}\left(\frac{\|\mathbf{v}\|_2}{\sqrt{n}}\prod_{l=1}^{L}\|\mathbf{W}_l\|_2\left[\sum_{l=1}^{L}\frac{\|\mathbf{W}_l^\top - \mathbf{W}_l^{(0)\top}\|_{2,1}^{2/3}}{\|\mathbf{W}_l\|_2^{2/3}}\right]^{3/2}\right)$$

or the bound given by (Neyshabur et al. 2018a)

$$\widetilde{O}\left(\frac{L\|\mathbf{v}\|_2}{\sqrt{n}}\prod_{l=1}^{L}\|\mathbf{W}_l\|_2\left[\sum_{l=1}^{L}\frac{(\sqrt{m}\|\mathbf{W}_l - \mathbf{W}_l^{(0)}\|_F)^2}{\|\mathbf{W}_l\|_2^2}\right]^{1/2}\right)$$

results in a generalization bound of the order $\widetilde{O}(m\tau/\sqrt{n})$. In comparison, when τ is small enough, our bound on the generalization gap is in the order of $\widetilde{O}(\tau\cdot\sqrt{m/n})$, which has a better dependency in m.

Theorem 3.1 in particular suggests that if gradient descent finds a parameter configuration with small surrogate error in $\mathcal{W}_{Rm^{-1/2}}$ for some R independent of m, then the obtained neural network has a generalization bound *decreasing* in m. The following lemma shows that under a gradient lower bound assumption, gradient descent indeed converges to a global minima in $\mathcal{W}_{Rm^{-1/2}}$ with R independent of m.

Theorem 3.2. *Suppose that the training loss function $L_S(\mathbf{W})$ satisfies the following inequality*

$$\left\|\nabla_{\mathbf{W}_L} L_S(\mathbf{W})\right\|_F \geqslant B\sqrt{m}\cdot\mathcal{E}_S(\mathbf{W}) \tag{3.1}$$

for all $\mathbf{W} \in \mathcal{W}_\tau$, where B is independent of m, and $\tau = \widetilde{O}(B^{-1}\epsilon^{-1}m^{-1/2})$. For any $\epsilon, \delta > 0$, there exist absolute constants $\overline{C}, \underline{C}$ and $m^* = \widetilde{O}(L^{12}B^{-4}\epsilon^{-2})\cdot\log(1/\delta)$ such that, if $m \geqslant m^*$, then with probability at least $1-\delta$, Algorithm 1 with step size $\eta = O(L^{-3}B^2m^{-1})$ generates $K = \widetilde{O}(L^3B^{-4}\epsilon^{-2})$ iterates $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(K)}$ that satisfy:

(i) $\mathbf{W}^{(k)} \in \mathcal{W}_\tau$, $k \in [K]$.

(ii) There exists $k \in \{0, \ldots, K-1\}$ such that $\mathcal{E}_S(\mathbf{W}^{(k)}) \leqslant \epsilon$.

Remark 3.2. *The gradient lower bound assumption* (3.1) *is by no means an unrealistic assumption. In fact, this assumption has been verified by several papers (Allen-Zhu, Li, and Song 2019; Zou et al. 2019; Zou and Gu 2019) under the assumption that $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \geqslant \phi$ for all $i, j \in [n]$, where $\phi > 0$ is an absolute constant. The corresponding value of B under this assumption is $\Omega(\mathrm{poly}(\phi, n^{-1}))$.*

Combining Theorems 3.1 and 3.2 directly gives the following corollary:

Corollary 3.1. *Suppose that the training loss function $L_S(\mathbf{W})$ satisfies inequality* (3.1) *for all $\mathbf{W} \in \mathcal{W}_\tau$, where B is independent of m, and $\tau = \widetilde{O}(B^{-1}\epsilon^{-1}m^{-1/2})$. For any $\epsilon, \delta > 0$, there exist absolute constants $\overline{C}, \underline{C}$ and $m^* = \widetilde{O}(L^{12}B^{-4}\epsilon^{-2})\cdot\log(1/\delta)$ such that, if $m \geqslant m^*$, then with probability at least $1-\delta$, Algorithm 1 with step size $\eta = O(L^{-3}B^2m^{-1})$ finds a point $\mathbf{W}^{(k)}$ that satisfies*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big[y\cdot f_{\mathbf{W}}(\mathbf{x}) < 0\big] \leqslant \epsilon + \widetilde{O}\big(L^2B^{-1}\epsilon^{-1}\cdot n^{-1/2} + L^4B^{-4/3}\epsilon^{-4/3}m^{-1/6}\big)$$

within $K = \widetilde{O}(L^3B^{-4}\epsilon^{-2})$ iterations.

As we discussed in Remark 3.2, if the pairwise distance between training inputs can be lower bounded by a constant ϕ, then (3.1) holds with $B = O(\mathrm{poly}(\phi, n^{-1}))$. However, plugging this value of B into the population error bound in Corollary 3.1 will give a bound $\widetilde{O}(\mathrm{poly}(n)\cdot n^{-1/2})$ (when m is large enough) which is vacuous and does not decrease in sample size n. We remark that this result is natural, because $B = \Omega(\mathrm{poly}(\phi, n^{-1}))$ corresponds to the condition that data inputs are separated, and in fact no assumption on the distribution of labels is made through out our analysis. Suppose that the labels are simply Rademacher variables and are independent of inputs, then clearly the expected error of any classifier cannot go below $1/2$, no matter how many training samples are used to learn the classifier. In the next subsection, we study particular data distribution assumptions under which (3.1) holds with a B that is not only independent of m, but also independent of n.

3.2 Generalization Error Bounds under Specific Data Distribution Assumptions

In this section we introduce two specific data distributions that have been studied in the literature, and show that if one of them holds, then (3.1) holds with a B independent of m and n. Assumption 3.3 below is related to a random feature model studied in (Rahimi and Recht 2009).

Assumption 3.3 (Separable by Random ReLU Feature)**.** *Denote by $p(\overline{\mathbf{u}})$ the density of standard Gaussian random vectors. Define*

$$\mathcal{F} = \left\{f(\mathbf{x}_i) = \int_{\mathbb{R}^d} c(\overline{\mathbf{u}})\sigma(\overline{\mathbf{u}}^\top\mathbf{x}_i)p(\overline{\mathbf{u}})\mathrm{d}\overline{\mathbf{u}} : \|c(\cdot)\|_\infty \leqslant 1\right\}.$$

We assume that there exist an $f(\cdot) \in \mathcal{F}$ and a constant $\gamma > 0$ such that $y_i\cdot f(\mathbf{x}_i) \geqslant \gamma$ for all $i \in [n]$.

$\mathcal{F}$ defined in Assumption 3.3 corresponds to the random feature function class studied in (Rahimi and Recht 2009) when the feature function is chosen to be ReLU. Assumption 3.3 essentially states that there exists a function f in the function class $\mathcal{F}$ that can separate the data distribution $\mathcal{D}$ with a constant margin γ. According to the definition of $\mathcal{F}$, each value of $\overline{\mathbf{u}}$ can be considered as a node in an infinite-width one-hidden-layer ReLU network, and the corresponding product $c(\overline{\mathbf{u}})p(\overline{\mathbf{u}})$ can be considered as the second-layer weight. Therefore $\mathcal{F}$ contains infinite-width one-hidden-layer ReLU networks whose second-layer weights decay faster than $p(\overline{\mathbf{u}})$. Also note that Assumption 3.3 is strictly milder than linearly separable assumption.

The following corollary gives an expected error bound of neural networks trained by gradient descent under Assumption 3.3.

Corollary 3.2. *Under Assumption 3.3, for any $\epsilon, \delta > 0$, there exist*

$$m^*(\epsilon, L, \gamma, \delta) = \widetilde{O}(\mathrm{poly}(2^L, \gamma^{-1})) \cdot \epsilon^{-14} \cdot \log(1/\delta),$$
$$n^*(\epsilon, L, \gamma, \delta) = \widetilde{O}(\mathrm{poly}(2^L, \gamma^{-1})) \cdot \epsilon^{-4} \cdot \log(1/\delta)$$

such that, if $m \geqslant m^*(\epsilon, L, \gamma, \delta)$ and $n \geqslant n^*(\epsilon, L, \gamma, \delta)$, then with probability at least $1-\delta$, Algorithm 1 with step size $\eta = O(4^{-L} L^{-3} \gamma^2 m^{-1})$ finds a point $\mathbf{W}^{(k)}$ that satisfies

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big[y \cdot f_{\mathbf{W}^{(k)}}(\mathbf{x}) > 0\big] \geqslant 1 - \epsilon$$

within $K = \widetilde{O}(\mathrm{poly}(2^L, \gamma^{-1})) \cdot \epsilon^{-2}$ iterations.

We now introduce another data distribution assumption which has been made in (Daniely 2017).

Assumption 3.4 (Separable by Conjugate Kernel)**.** *The conjugate kernel of fully connected neural networks is defined recursively as*

$$\kappa^{(0)}(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle,$$
$$\kappa^{(l+1)}(\mathbf{x}, \mathbf{x}') = \mathbb{E}_{f\sim N(\mathbf{0}, \kappa^{(l)})}[\sigma(f(\mathbf{x}))\sigma(f(\mathbf{x}'))].$$

We assume that there exists a function f in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ induced by the conjugate kernel function $\kappa^{(L-1)}(\cdot,\cdot)$ with $\|f\|_{\mathcal{H}} \leqslant 1$ such that $y_i \cdot f(\mathbf{x}_i) \geqslant \gamma > 0$.

Under Assumption 3.4, we have the following result.

Corollary 3.3. *Under Assumption 3.4, for any $\epsilon, \delta > 0$, there exist*

$$m^*(\epsilon, L, \gamma, \delta) = \widetilde{O}(\mathrm{poly}(L, \gamma^{-1})) \cdot \epsilon^{-14} \cdot \log(1/\delta),$$
$$n^*(\epsilon, L, \gamma, \delta) = \widetilde{O}(\mathrm{poly}(L, \gamma^{-1})) \cdot \epsilon^{-4} \cdot \log(1/\delta)$$

such that, if $m \geqslant m^*(\epsilon, L, \gamma, \delta)$ and $n \geqslant n^*(\epsilon, L, \gamma, \delta)$, then with probability at least $1-\delta$, Algorithm 1 with step size $\eta = O(4^{-L} L^{-3} \gamma^2 m^{-1})$ finds a point $\mathbf{W}^{(k)}$ that satisfies

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big[y \cdot f_{\mathbf{W}^{(k)}}(\mathbf{x}) > 0\big] \geqslant 1 - \epsilon$$

within $K = \widetilde{O}(\mathrm{poly}(\gamma^{-1})) \cdot \epsilon^{-2}$ iterations.

Remark 3.3. *Corollary 3.3 shows that under Assumption 3.4, a neural network trained by gradient descent can achieve ϵ-expected error given $\widetilde{O}(\epsilon^{-4})$ training examples. We remark that although (Daniely 2017) studied the same assumption, our result is not a re-derivation of the results given by (Daniely 2017), because while they considered one-pass SGD and square loss, while we consider GD and cross-entropy loss. More importantly, Assumption 3.4 is just one specific setting our result can cover, and therefore Corollary 3.3 demonstrates the power of our general theory.*

Remark 3.4. *A follow-up work (Cao and Gu 2019) studied the generalization performance of over-parameterized neural networks trained with one-pass SGD, and relate the generalization bound to the neural tangent kernel function studied in recent work (Jacot, Gabriel, and Hongler 2018). We remark that their generalization bound is based on an online-to-batch conversion argument, which cannot be applied to the standard gradient descent algorithm we study in this paper. Therefore our result and their result are not directly comparible.*

4 Proof of the Main Theory

In this section we provide the proofs of the main results given in Section 3. The omitted proof can be found in the supplementary material.

4.1 Proof of Theorem 3.1

Here we provide the detailed proof of Theorem 3.1. We first present the lemma below, which gives an upper bound on the gradients of $L_S(\mathbf{W})$, and relates the gradients with the empirical surrogate error $\mathcal{E}_S(\mathbf{W})$.

Lemma 4.1. *For any $\delta > 0$, if*

$$m \geqslant \overline{C} \max\{L^2 \log(mn/\delta), L^{-8/3} \tau^{-4/3} \log[m/(\tau\delta)]\},$$
$$\tau \leqslant \underline{C} L^{-6} [\log(m)]^{-3/2}$$

for some large enough absolute constant $\overline{C}$ and small enough absolute constant $\underline{C}$, then with probability at least $1-\delta$, for all $\mathbf{W} \in \mathcal{W}_\tau$ and $l \in [L]$,

$$\big\|\nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\big\|_F \leqslant C\sqrt{m},$$
$$\big\|\nabla_{\mathbf{W}_l} L_S(\mathbf{W})\big\|_F \leqslant C\sqrt{m} \cdot \mathcal{E}_S(\mathbf{W}),$$

where C is an absolute constant.

Lemma 4.2 below reveals the fact that near initialization, the neural network function is *almost linear* in terms of its weight parameters. As a consequence, the empirical loss function $L_S(\mathbf{W})$ is *almost smooth* in a small neighborhood around $\mathbf{W}^{(0)}$.

Lemma 4.2. *For any $\delta > 0$, if*

$$m \geqslant \overline{C} \max\{L^2 \log(mn/\delta), L^{-8/3} \tau^{-4/3} \log[m/(\tau\delta)]\},$$
$$\tau \leqslant \underline{C} L^{-6} [\log(m)]^{-3/2}$$

for some large enough absolute constant $\overline{C}$ and small enough absolute constant $\underline{C}$, then there exists an absolute constant C such that with probability at least $1-\delta$, for all $\widetilde{\mathbf{W}}, \widehat{\mathbf{W}} \in \mathcal{W}_\tau$,

$$\big|f_{\widehat{\mathbf{W}}}(\boldsymbol{x}_i) - F_{\widetilde{\mathbf{W}}, \widehat{\mathbf{W}}}(\boldsymbol{x}_i)\big| \leqslant C L^2 \tau^{1/3} \sqrt{m \log(m)} \cdot \sum_{l=1}^{L} \big\|\widehat{\mathbf{W}}_l - \widetilde{\mathbf{W}}_l\big\|_2,$$

where

$$F_{\widetilde{\mathbf{W}}, \widehat{\mathbf{W}}}(\mathbf{x}) = f_{\widetilde{\mathbf{W}}}(\mathbf{x}) + \sum_{l=1}^{L} \mathrm{Tr}\big[(\widehat{\mathbf{W}}_l - \widetilde{\mathbf{W}}_l)^\top \nabla_{\mathbf{W}_l} f_{\widetilde{\mathbf{W}}}(\mathbf{x})\big],$$

and

$$\begin{aligned} L_S(\widehat{\mathbf{W}}) - L_S(\widetilde{\mathbf{W}}) &\leqslant C \sum_{l=1}^{L} L^2 \tau^{1/3} \sqrt{m \log(m)} \cdot \big\|\widehat{\mathbf{W}}_l - \widetilde{\mathbf{W}}_l\big\|_2 \cdot \mathcal{E}_S(\widetilde{\mathbf{W}}) \\ &\quad + \sum_{l=1}^{L} \mathrm{Tr}[(\widehat{\mathbf{W}}_l - \widetilde{\mathbf{W}}_l)^\top \nabla_{W_l} L_S(\widetilde{\mathbf{W}})] \\ &\quad + C \sum_{l=1}^{L} m L^3 \cdot \|\widehat{\mathbf{W}}_l - \widetilde{\mathbf{W}}_l\|_2^2. \end{aligned}$$

Proof of Theorem 3.1. Let $\mathcal{F}_\tau = \{f_{\mathbf{W}}(\mathbf{x}) : \mathbf{W} \in \mathcal{W}_\tau\}$. We consider the empirical Rademacher complexity (Bartlett and Mendelson 2002; Mohri, Rostamizadeh, and Talwalkar 2018; Shalev-Shwartz and Ben-David 2014) of $\mathcal{F}_\tau$ defined as follows

$$\widehat{\mathfrak{R}}_n[\mathcal{F}_\tau] = \mathbb{E}_{\boldsymbol{\xi}}\bigg[\sup_{\mathbf{W}\in\mathcal{W}_\tau} \frac{1}{n}\sum_{i=1}^n \xi_i f_{\mathbf{W}}(\boldsymbol{x}_i)\bigg],$$

where $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_n)^\top$ is an n-dimensional vector consisting of independent Rademacher random variables $\xi_1,\ldots,\xi_n$. Since $y \in \{+1,1\}$, $|\ell'(z)| \leqslant 1$ and $\ell'(z)$ is 1-Lipschitz continuous, by symmetrization and the standard uniform convergence results in terms of empirical Rademacher complexity (Mohri, Rostamizadeh, and Talwalkar 2018; Shalev-Shwartz and Ben-David 2014), with probability at least $1-\delta$ we have

$$\begin{aligned}
&\sup_{\mathbf{W}\in\mathcal{W}_\tau} |\mathcal{E}_S(\mathbf{W}) - \mathcal{E}_{\mathcal{D}}(\mathbf{W})| \\
&= \sup_{\mathbf{W}\in\mathcal{W}_\tau} \bigg|\frac{1}{n}\sum_{i=1}^n \ell'\big[y_i\cdot f_{\mathbf{W}}(\boldsymbol{x}_i)\big] - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\ell'\big[y\cdot f_{\mathbf{W}}(\mathbf{x})\big]\bigg| \\
&\leqslant 2\widehat{\mathfrak{R}}_n[\mathcal{F}_\tau] + C_1\sqrt{\frac{\log(1/\delta)}{n}},
\end{aligned}$$

where C_1 is an absolute constant. We now bound the term $\widehat{\mathfrak{R}}_n[\mathcal{F}_\tau]$. By definition, we have

$$\widehat{\mathfrak{R}}_n[\mathcal{F}_\tau] \leqslant I_1 + I_2, \tag{4.1}$$

where

$$\begin{aligned}
I_1 &= \mathbb{E}_{\boldsymbol{\xi}}\bigg\{\sup_{\mathbf{W}\in\mathcal{W}_\tau} \frac{1}{n}\sum_{i=1}^n \xi_i\big[f_{\mathbf{W}}(\boldsymbol{x}_i) - F_{\mathbf{W}^{(0)},\mathbf{W}}(\boldsymbol{x}_i)\big]\bigg\}, \\
I_2 &= \mathbb{E}_{\boldsymbol{\xi}}\bigg\{\sup_{\mathbf{W}\in\mathcal{W}_\tau} \frac{1}{n}\sum_{i=1}^n \xi_i F_{\mathbf{W}^{(0)},\mathbf{W}}(\boldsymbol{x}_i)\bigg\},
\end{aligned}$$

and

$$\begin{aligned}
F_{\mathbf{W}^{(0)},\mathbf{W}}(\mathbf{x}) = \sum_{l=1}^L \mathrm{Tr}\big[(\mathbf{W}_l - \mathbf{W}_l^{(0)})^\top \nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\mathbf{x})\big] \\
+ f_{\mathbf{W}^{(0)}}(\mathbf{x}).
\end{aligned}$$

For I_1, by Lemma 4.2, we have

$$\begin{aligned}
I_1 &\leqslant \max_{i\in[n]} \big|f_{\mathbf{W}}(\boldsymbol{x}_i) - F_{\mathbf{W}^{(0)},\mathbf{W}}(\boldsymbol{x}_i)\big| \\
&\leqslant C_2 L^4\sqrt{m\log(m)}\tau^{4/3}
\end{aligned}$$

for all $i\in[n]$, where C_2 is an absolute constant. For I_2, note that $\mathbb{E}_{\boldsymbol{\xi}}\big\{\sup_{\mathbf{W}\in\mathcal{W}_\tau}\sum_{i=1}^n \xi_i f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\big\} = 0$, and therefore

$$\begin{aligned}
I_2 &= \frac{1}{n}\sum_{l=1}^L \mathbb{E}_{\boldsymbol{\xi}}\bigg\{\sup_{\|\widetilde{\mathbf{W}}_l\|_F\leqslant\tau} \mathrm{Tr}\bigg[\widetilde{\mathbf{W}}_l^\top \sum_{i=1}^n \xi_i \nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\bigg]\bigg\} \\
&\leqslant \frac{\tau}{n}\sum_{l=1}^L \mathbb{E}_{\boldsymbol{\xi}}\bigg[\bigg\|\sum_{i=1}^n \xi_i \nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\bigg\|_F\bigg].
\end{aligned}$$

By Jensen's inequality,

$$\begin{aligned}
I_2 &\leqslant \frac{\tau}{n}\sum_{l=1}^L \sqrt{\mathbb{E}_{\boldsymbol{\xi}}\bigg[\bigg\|\sum_{i=1}^n \xi_i \nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\bigg\|_F^2\bigg]} \\
&= \frac{\tau}{n}\sum_{l=1}^L \sqrt{\sum_{i=1}^n \big\|\nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\big\|_F^2}.
\end{aligned}$$

Now by Lemma 4.1, we have $\big\|\nabla_{\mathbf{W}_l} f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)\big\|_F \leqslant C_3\sqrt{m}$ for all $l\in[L]$, where C_3 is an absolute constant. Therefore $I_2 \leqslant C_3 L\tau\cdot\sqrt{m/n}$. Plugging in the bounds of I_1 and I_2 into (4.1) and applying Markov's inequality

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\big\{-\ell'\big[y\cdot f_{\mathbf{W}}(\mathbf{x})\big]\big\} \\
&\quad\geqslant \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big\{-\ell'\big[y\cdot f_{\mathbf{W}}(\mathbf{x})\big] \geqslant 1/2\big\}/2 \\
&\quad= \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\big[y\cdot f_{\mathbf{W}}(\mathbf{x}) < 0\big]/2
\end{aligned}$$

completes the proof. □

4.2 Proof of Theorem 3.2

The following lemma is given by (Zou et al. 2019), which gives a bound on the neural network output at initialization.

Lemma 4.3 ((Zou et al. 2019))**.** *For any $\delta > 0$, with probability at least $1-\delta$, $|f_{\mathbf{W}^{(0)}}(\boldsymbol{x}_i)| \leqslant C\sqrt{\log(n/\delta)}$ for all $i\in[n]$, where C is an absolute constant.*

Proof of Theorem 3.2. Set $\tau = \widetilde{O}(B^{-1}\epsilon^{-1}m^{-1/2})$. Then there exist $\eta = O(L^{-3}B^2m^{-1})$, $K = \widetilde{O}(L^3B^{-4}\epsilon^{-2})$ and $m^* = \widetilde{O}(L^{12}B^{-4}\epsilon^{-2})\cdot\log(1/\delta)$ such that when $m \geqslant m^*$, all assumptions of Lemmas 4.1, 4.2 hold, and

$$(K\eta)^{1/2}B^{-1}[\log(n/\delta)]^{1/2} \leqslant \nu\tau, \tag{4.2}$$

$$L^3m\eta \leqslant \nu B^2, \tag{4.3}$$

$$L^2\tau^{1/3}[m\log(m)]^{1/2} \leqslant \nu B^2 m \tag{4.4}$$

$$(K\eta\cdot m)^{-1/2}B^{-1} \leqslant \epsilon \tag{4.5}$$

for some small enough absolute constant ν. We now prove by induction that $\mathbf{W}^{(k)} \in \mathcal{W}(\mathbf{W}^{(0)},\tau/2)$, $k\in\{0\}\cup[K]$. By definition clearly we have $\mathbf{W}^{(0)} \in \mathcal{W}(\mathbf{W}^{(0)},\tau/2)$. Suppose that $\mathbf{W}^{(k)} \in \mathcal{W}(\mathbf{W}^{(0)},\tau/2)$ for all $k = 0,\ldots,t$. Then for all $l\in[L]$ we have

$$\begin{aligned}
&\|\mathbf{W}_l^{(t+1)} - \mathbf{W}_l^{(0)}\|_F \\
&\quad\leqslant \|\mathbf{W}_l^{(t)} - \mathbf{W}_l^{(0)}\|_F + \eta\|\nabla_{\mathbf{W}_l} L_S(\mathbf{W}^{(t)})\|_F \\
&\quad\leqslant \tau/2 + \tau/2 = \tau,
\end{aligned}$$

where the last inequality follows by Lemma 4.1 and the definition of τ and η (note that a comparison between (3.1) and Lemma 4.1 implies that $B = O(1)$). Therefore $\mathbf{W}^{(t+1)} \in \mathcal{W}_\tau$. Plugging in the gradient upper bound given by Lemma 4.1 and assumption (3.1) into the result of Lemma 4.2, we obtain

$$\begin{aligned}
&L_S(\mathbf{W}^{(k+1)}) - L_S(\mathbf{W}^{(k)}) \\
&\leqslant C_1\sum_{l=1}^L \big[L^2\tau^{1/3}\eta m\sqrt{\log(m)} + L^3m^2\eta^2\big]\cdot\mathcal{E}_S^2(\mathbf{W}^{(k)}) \\
&\quad - \eta\cdot B^2 m\cdot\mathcal{E}_S^2(\mathbf{W}^{(k)})
\end{aligned}$$

for all $k = 0, \ldots, t$, where C_1, C_2 are absolute constants. Plugging in the bounds (4.3) and (4.4), we have

$$L_S(\mathbf{W}^{(k+1)}) - L_S(\mathbf{W}^{(k)}) \leqslant -\eta B^2 m \mathcal{E}_S^2(\mathbf{W}^{(k)})/2 \quad (4.6)$$

for all $k = 0, \ldots, t$. Combining (4.6) with Lemma 4.1 gives

$$\begin{aligned}&\big\|\nabla_{\mathbf{W}_l} L_S(\mathbf{W}^{(k)})\big\|_F \\ &\quad\leqslant C_2 \eta^{-1/2} B^{-1} [L_S(\mathbf{W}^{(k)}) - L_S(\mathbf{W}^{(k+1)})]^{1/2}\end{aligned}$$

for all $k = 0, \ldots, t$, where C_2 is an absolute constant. Note that by Lemma 4.3 and the fact that $\ell(z) \leqslant 1 + |z|$, we have $L_S(\mathbf{W}^{(0)}) - L_S(\mathbf{W}^{(K)}) \leqslant C_3 [\log(n/\delta)]^{1/2}$ for some absolute constant C_3. Therefore by Jensen's inequality,

$$\begin{aligned}&\|\mathbf{W}_l^{(t+1)} - \mathbf{W}_l^{(0)}\|_F \\ &\leqslant \eta \sum_{k=0}^{t} \big\|\nabla_{\mathbf{W}_l} L_S(\mathbf{W}^{(k)})\big\|_F \\ &\leqslant C_2 \eta^{1/2} B^{-1} \sum_{k=0}^{t} [L_S(\mathbf{W}^{(k)}) - L_S(\mathbf{W}^{(k+1)})]^{1/2} \\ &\leqslant C_2 \sqrt{K\eta} B^{-1} \cdot [L_S(\mathbf{W}^{(0)}) - L_S(\mathbf{W}^{(k)})]^{1/2} \\ &\leqslant C_4 \sqrt{K\eta} B^{-1} [\log(n/\delta)]^{1/2} \\ &\leqslant \tau/2,\end{aligned}$$

where C_4 is an absolute constant, and the last inequality follows by (4.2). Therefore by induction, $\mathbf{W}^{(k)} \in \mathcal{W}(\mathbf{W}^{(0)}, \tau/2)$ for all $k \in [K]$. This also implies that (4.6) holds for all $k = 0, \ldots, K-1$. Let $k^* = \mathrm{argmin}_{k \in \{0, \ldots, K-1\}} \mathcal{E}_S(\mathbf{W}^{(k)})$. Telescoping over k gives

$$L_S(\mathbf{W}^{(K)}) - L_S(\mathbf{W}^{(0)}) \leqslant -K\eta B^2 m \cdot \mathcal{E}_S^2(\mathbf{W}^{(k^*)}).$$

Hence by (4.5) we have

$$\mathcal{E}_S(\mathbf{W}^{(k^*)}) \leqslant (K\eta \cdot m)^{-1/2} B^{-1} \leqslant \epsilon,$$

This completes the proof. □

4.3 Proof of Corollary 3.2

In this section we give the proof of Corollary 3.2. The following lemma verifies that under Assumption 3.3, (3.1) indeed holds with B independent in both m and n.

Lemma 4.4. *For any $\delta > 0$, if*

$$\begin{aligned}m &\geqslant \overline{C} \cdot \max\{4^L L^2 \gamma^{-2} \log(mnL/\delta), \\ &\qquad L^{-8/3} \tau^{-4/3} \log[m/(\tau\delta)]\}, \\ \tau &\leqslant \underline{C} \cdot 8^{-L} L^{-2} \gamma^3 [\log(m)]^{-3/2}\end{aligned}$$

for some large enough absolute constant $\overline{C}$ and small enough absolute constant $\underline{C}$, then with probability at least $1 - \delta$, there exists an absolute constant C such that

$$\big\|\nabla_{\mathbf{W}_L} L_S(\mathbf{W})\big\|_F \geqslant C \cdot 2^{-L} \cdot \gamma \sqrt{m} \cdot \mathcal{E}_S(\mathbf{W})$$

for all $\mathbf{W} \in \mathcal{W}_\tau$.

Proof of Corollary 3.2. Corollary 3.2 directly follows by plugging in $B = O(2^{-L}\gamma)$ given by Lemma 4.4 and the assumptions $m \geqslant \widetilde{O}(L^{24} 2^{8L} \gamma^{-8}) \cdot \epsilon^{-14}$, $n \geqslant \widetilde{O}(L^4 4^L \gamma^{-2}) \cdot \epsilon^{-4}$ into Corollary 3.1. □

4.4 Proof of Corollary 3.3

In this section we give the proof of Corollary 3.3. Similar to the proof of Corollary 3.2, we mainly need to derive a gradient lower bound of the form (3.1). The result is given in the following lemma, which gives a similar result in part of the proof of Claim 1 in (Daniely 2017).

Lemma 4.5. *For any $\delta > 0$, if*

$$\begin{aligned}&m \geqslant \overline{C} \cdot \max\{\gamma^{-2} \log(mn/\delta), \tau^{-4/3} \log[m/(\tau\delta)]\}, \\ &\tau \leqslant \underline{C} \cdot \gamma^3 [\log(m)]^{-3/2}\end{aligned}$$

for some large enough absolute constant $\overline{C}$ and small enough absolute constant $\underline{C}$, then with probability at least $1 - \delta$, there exists an absolute constant C such that

$$\big\|\nabla_{\mathbf{W}_L} L_S(\mathbf{W})\big\|_F \geqslant C\gamma\sqrt{m} \cdot \mathcal{E}_S(\mathbf{W})$$

for all $\mathbf{W} \in \mathcal{W}_\tau$.

Proof of Corollary 3.3. Corollary 3.3 directly follows by plugging in $B = O(\gamma)$ given by Lemma 4.5 and the assumptions $m \geqslant \widetilde{O}(L^{24}\gamma^{-8}) \cdot \epsilon^{-14}$, $n \geqslant \widetilde{O}(L^4\gamma^{-2}) \cdot \epsilon^{-4}$ into Corollary 3.1. □

5 Conclusions and Future Work

In this paper, we provided a generalization guarantee of gradient descent for training deep ReLU networks under over-parameterization, which hold under mild data distribution assumptions. Although we only focus on gradient descent and cross-entropy loss for binary classification, our results can be extended to stochastic gradient descent, other loss functions and multi-class classification. In addition, we will derive generalization bounds for deep learning based on the "small-ball" assumption proposed in (Mendelson 2014). Another interesting direction is to investigate the generalization of gradient descent using stability-based analysis (Hardt, Recht, and Singer 2016).

Acknowledgements

We thank the anonymous reviewers and senior PC for their helpful comments. This research was sponsored in part by the National Science Foundation CAREER Award IIS-1906169, IIS-1903202, and Salesforce Deep Learning Research Award. The views and conclusions contained in this paper are those of the authors.

References

Allen-Zhu, Z.; Li, Y.; and Liang, Y. 2019. Learning and generalization in overparameterized neural networks, going beyond two layers. In *NeurIPS*.

Allen-Zhu, Z.; Li, Y.; and Song, Z. 2019. A convergence theory for deep learning via over-parameterization. In *ICML*.

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

Arora, S.; Ge, R.; Neyshabur, B.; and Zhang, Y. 2018b. Stronger generalization bounds for deep nets via a compression approach. In *ICML*, 254–263.

Arora, S.; Du, S.; Hu, W.; Li, Z.; and Wang, R. 2019. Fine-grained analysis of optimization and generalization for over-parameterized two-layer neural networks. In *ICML*.

Arora, S.; Cohen, N.; and Hazan, E. 2018. On the optimization of deep networks: Implicit acceleration by overparameterization. In *ICML*, 244–253.

Bartlett, P. L., and Mendelson, S. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3(Nov):463–482.

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. In *NIPS*, 6241–6250.

Brutzkus, A.; Globerson, A.; Malach, E.; and Shalev-Shwartz, S. 2018. Sgd learns over-parameterized networks that provably generalize on linearly separable data. In *ICLR*.

Cao, Y., and Gu, Q. 2019. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *NeurIPS*.

Daniely, A. 2017. Sgd learns the conjugate kernel class of the network. In *NIPS*, 2422–2430.

Du, S. S.; Lee, J. D.; Tian, Y.; Singh, A.; and Poczos, B. 2018. Gradient descent learns one-hidden-layer cnn: Dont be afraid of spurious local minima. In *ICML*, 1338–1347.

Du, S.; Lee, J.; Li, H.; Wang, L.; and Zhai, X. 2019a. Gradient descent finds global minima of deep neural networks. In *ICML*, 1675–1685.

Du, S. S.; Zhai, X.; Poczos, B.; and Singh, A. 2019b. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*.

Dziugaite, G. K., and Roy, D. M. 2017. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *UAI*.

Golowich, N.; Rakhlin, A.; and Shamir, O. 2018. Size-independent sample complexity of neural networks. In *COLT*, 297–299.

Haeffele, B. D., and Vidal, R. 2015. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*.

Hardt, M.; Recht, B.; and Singer, Y. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *ICML*, 1225–1234.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29(6):82–97.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *NIPS*, 8571–8580.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Li, Y., and Liang, Y. 2018. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *NIPS*, 8157–8166.

Li, X.; Lu, J.; Wang, Z.; Haupt, J.; and Zhao, T. 2018. On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*.

Mendelson, S. 2014. Learning without concentration. In *COLT*, 25–39.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of machine learning*. MIT press.

Neyshabur, B.; Bhojanapalli, S.; McAllester, D.; and Srebro, N. 2018a. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *ICLR*.

Neyshabur, B.; Li, Z.; Bhojanapalli, S.; LeCun, Y.; and Srebro, N. 2018b. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*.

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2015. Norm-based capacity control in neural networks. In *COLT*.

Nguyen, Q., and Hein, M. 2017. The loss surface of deep and wide neural networks. In *ICML*, 2603–2612.

Rahimi, A., and Recht, B. 2009. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NIPS*, 1313–1320.

Safran, I., and Shamir, O. 2016. On the quality of the initial basin in overspecified neural networks. In *ICML*, 774–782.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Song, M.; Montanari, A.; and Nguyen, P. 2018. A mean field view of the landscape of two-layers neural networks. *PNAS* 115:E7665–E7671.

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

Wei, C.; Lee, J. D.; Liu, Q.; and Ma, T. 2019. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. *NeurIPS*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zou, D., and Gu, Q. 2019. An improved analysis of training over-parameterized deep neural networks. In *NeurIPS*.

Zou, D.; Cao, Y.; Zhou, D.; and Gu, Q. 2019. Stochastic gradient descent optimizes over-parameterized deep ReLU networks. In *Machine Learning Journal*.