Gradient Descent Maximizes the Margin of Homogeneous Neural Networks

Kaifeng Lyu * Jian Li †

Abstract

Recent works on implicit regularization have shown that gradient descent converges to the max-margin direction for logistic regression with one-layer or multi-layer linear networks. In this paper, we generalize this result to homogeneous neural networks, including fully-connected and convolutional neural networks with ReLU or LeakyReLU activations. In particular, we study the gradient flow (gradient descent with infinitesimal step size) optimizing the logistic loss or cross-entropy loss of any homogeneous model (possibly non-smooth), and show that if the training loss decreases below a certain threshold, then we can define a smoothed version of the normalized margin which increases over time. We also formulate a natural constrained optimization problem related to margin maximization, and prove that both the normalized margin and its smoothed version converge to the objective value at a KKT point of the optimization problem. Furthermore, we extend the above results to a large family of loss functions. We conduct several experiments to justify our theoretical finding on MNIST and CIFAR-10 datasets. For gradient descent with constant learning rate, we observe that the normalized margin indeed keeps increasing after the dataset is fitted, but the speed is very slow. However, if we schedule the learning rate more carefully, we can observe a more rapid growth of the normalized margin. Finally, as margin is closely related to robustness, we discuss potential benefits of training longer for improving the robustness of the model.

1 Introduction

Despite that the loss landscape of a neural network is typically highly non-convex, in practice it is usually not difficult to achieve a very small training loss (say, 0.001) using gradient descent. For over-parameterized neural networks with very large width, several recent theoretical results also have justified this phenomenon (see e.g., [Du et al., 2019, Allen-Zhu et al., 2019, Zou et al., 2018]). However, as modern neural networks may have many global minima, it is still a major open question in deep learning why gradient descent or its variants, are biased towards solutions with good generalization performance on the test set. To achieve a better understanding of the question, previous works have studied the implicit regularization of gradient descent in different settings. Perhaps the simplest setting is linear logistic regression on linearly separable data. In this setting, the model is parameterized by a weight vector \(w\). A data point \(x\) is classified to be positive if \(w^\top x > 0\); otherwise \(x\) is classified to be negative. Therefore, only the direction \(w/\|w\|\) is important for making prediction. For showing implicit regularization effects of gradient descent, it is sufficient to characterize the convergent direction of \(w\). Soudry et al. [2018b],[a], Ji and Telgarsky [2018], Nacson et al. [2018a,b] investigated this problem and proved that the direction of \(w\) converges to the direction that maximizes the \(L^2\)-margin while the norm of \(w\) diverges to \(+\infty\), if training with (stochastic) gradient descent on logistic loss. Interestingly, this convergent direction is the same as that of any regularization path: any sequence of weight vectors \(\{w_t\}\) such that every \(w_t\) is a global minimum of the \(L^2\)-regularized loss \(L(w) + \frac{\lambda}{2} \|w\|_2^2\) with \(\lambda_i \to 0\) [Rosset et al., 2004]. Indeed, the trajectory of gradient descent is also pointwise close to a regularization path [Suggala et al., 2018].

The aforementioned linear logistic regression can be viewed as a single-layer neural network. A natural and important question is to what extent gradient descent has the same implicit regularization effect for modern deep neural networks. To answer this question, we first note a simple fact that many deep neural networks do not have finite global minima, since one can always obtain a smaller training loss by scaling up the homogeneous part of the parameters (after

*Institute for Interdisciplinary Information Sciences, Tsinghua University. E-mail: lkf15@mails.tsinghua.edu.cn
†Institute for Interdisciplinary Information Sciences, Tsinghua University. E-mail: lijian83@mail.tsinghua.edu.cn
model fits the training data perfectly). For example, given a Convolutional Neural Network (CNN) that has achieved 100% training accuracy, one can easily make the cross-entropy loss arbitrarily small by scaling up the weight and bias parameters \((W, b)\) at the last layer, i.e., transforming \((W, b)\) to \((cW, cb)\) for large enough \(c > 0\). This means that, similar to linear logistic regression, CNNs also have some parameters whose scale does not matter, and hence a promising and meaningful research direction is to study whether their convergent direction maximizes the margin. In general, we observe that the following three properties that are usually satisfied by modern deep neural networks:

1. **Partial Homogeneity.** The output of the neural network is (positively) homogeneous with respect to a part of its parameters (e.g., the parameters at the last linear layer);

2. **Separability.** The training set is separable by the neural network for some set of parameters, i.e., the neural network has sufficient representation power to achieve 100% training accuracy (this is true for state-of-the-art CNNs for image classification, and many of them even have enough capacity to fit randomly labeled data easily [Zhang et al., 2017]);

3. **No finite minima on the loss function.** The loss function used to measure the similarity between the network output and ground-truth is lower bounded by a constant (e.g., 0) but it does not have finite minima (e.g., exponential loss, logistic loss, cross-entropy loss).

For a deep neural network with the above properties, one can see that gradient descent may not converge to a point as the homogeneous part can grow to infinity (while the loss keeps decreasing). In light of this fact, it is meaningful to study the direction of those parameters and the corresponding margin.

It is shown in [Wei et al., 2018] that the regularization path does converge to the max-margin direction for (fully) homogeneous neural networks with cross-entropy or logistic loss. For gradient flow, similar results on the convergent direction are known for linear fully-connected networks [Ji and Telgarsky, 2019]. For gradient descent on linear fully-connected and convolutional networks, [Gunasekar et al., 2018b] formulates a constrained optimization problem and proves that gradient descent converges to the direction of a KKT point of this problem or even the max-margin direction, under various assumptions including the convergence of loss and gradient directions. In an independent work, [Nacson et al., 2019] generalize the result in [Gunasekar et al., 2018b] to smooth homogeneous models (we will discuss this work in more details in Section 2).

### 1.1 Main Results

In this paper, we aim to analyze the implicit regularization of gradient descent for deep homogeneous neural networks with a minimal set of assumptions: Homogeneity, Separability, No Finite Minima on the loss function. Recall that we have stated the second and third assumptions, and now we formally introduce the Homogeneity assumption.

**Homogeneity.** Motivated by the aforementioned results on regularization path [Rosset et al., 2004, Wei et al., 2018], we assume **Full Homogeneity** instead of Partial Homogeneity. That is, for a neural network parameterized by \(\theta\), its output \(\Phi(\theta; x)\), where \(x\) stands for the input, satisfies the following: there exists a number \(L\) (called the order) such that

\[
\forall c > 0 : \Phi(c\theta; x) = c^L \Phi(\theta; x) \quad \text{for all } \theta \text{ and } x.
\]

It is important to note that many neural networks satisfy Full Homogeneity [Neyshabur et al., 2015, Du et al., 2018]. For example, deep fully-connected neural networks or deep CNNs with ReLU or LeakyReLU activations can be made homogeneous if we remove all the bias terms, and the order \(L\) is exactly equal to the number of layers.

**Simplifications.** For simplicity and ease of presentation, we make the following simplifications. First, as the most prominent examples of homogeneous neural networks are all non-smooth (e.g., ReLU networks), we turn to analyze the case of training neural networks by gradient flow (more precisely, subgradient flow in Clarke’s sense).

Second, we ensure Separability as follows: we assume that after time \(t_0\), the training loss is smaller than a threshold, and the threshold here is chosen to be so small that the training accuracy is guaranteed to be 100% (e.g., for the logistic loss and cross-entropy loss, the threshold can be set to \(\ln 2\)). In this paper, we focus on analyzing the behavior of the network after \(t_0\).
Our Contribution. As we mentioned before, with homogeneity, the parameter $\theta$ may keep increasing and it makes sense to investigate the direction of $\theta$. For a fixed direction, one can see that the margin $\gamma(\theta)$ scales linearly with $\|\theta\|_2^2$. This motivates us to study the normalized margin, $\bar{\gamma}(\theta) := \gamma(\theta)/\|\theta\|_2^L$ (see also Section 3.1).

The main claim of our work is the following:

Gradient descent does maximize the normalized margin for homogeneous models.

In particular, our theoretical results can answer the following questions regarding the normalized margin.

(i). First, how does the normalized margin change during training? The answer may seem complicated since one can easily come up with examples in which $\bar{\gamma}$ increases or decreases in a short time interval. However, we can show that the overall tendency of the normalized margin is increasing in the following sense: for a large family of loss functions (including logistic loss and cross-entropy loss), there exists a smoothed version of the normalized margin, denoted as $\tilde{\gamma}$ (see Definition 4.1), such that (1) $|\tilde{\gamma} - \bar{\gamma}| \to 0$ as $t \to \infty$; and (2) $\tilde{\gamma}$ is non-decreasing for $t > t_0$ (recall that $t_0$ is a time that the training loss is less than the threshold). See Theorem 4.2 (for exponential loss) and Theorem A.1 (for a more general family of loss functions). Furthermore, we provide asymptotically tight convergence/growth rate of the loss and the weights (see Theorem F.1) during the training for $t > t_0$.

(ii). Second, how large is the normalized margin at convergence? To answer this question, we formulate a natural constrained optimization problem which aims to directly maximize the margin. However, since the objective is highly non-convex and non-smooth, we cannot hope for a globally optimal margin at convergence. In fact, even to show that the convergent margin is a local optimum may not be possible, as gradient flow may get stuck at saddle points if we do not inject any noise [Du et al. 2017]. In this work, we show the best that we can hope for in our setting: every limit point of $\{\theta(t)/\|\theta(t)\|_2 : t > 0\}$ is along the direction of a KKT point of the max-margin problem. See Theorem 4.2 (for exponential loss) and Theorem C.4 (for a more general family of loss functions). This result can be seen as a significant generalization of previous works [Soudry et al. 2018a,b, Ji and Telgarsky 2019, Gunasekar et al. 2018b] from linear classifiers to homogeneous classifiers.

Experiments. The main practical implication of our theoretical result is that training longer can enlarge the normalized margin. To justify this claim empirically, we train CNNs on MNIST (see Section 7.1) and CIFAR-10 (see Appendix H.4) with SGD. For constant step size, we can see that the normalized margin keeps increasing, but the growth rate is rather slow (because the gradient gets smaller and smaller). To speed up the training, we propose a learning rate scheduling method which enlarges the learning rate according to the current training loss. With the new learning rate scheduling, we can see the loss decreases exponentially faster and the normalized margin increases significantly faster as well.

For feedforward neural networks with ReLU activation, the normalized margin on a training sample is closely related to the $L^2$-robustness (the $L^2$-distance from the training sample to the decision boundary). Indeed, the former divided by a Lipschitz constant is a lower bound for the latter (See Section 7.2). For example, the normalized margin is a lower bound for the $L^2$-robustness on fully-connected networks with ReLU activation (See, e.g., Theorem 4 in [Sokolic et al. 2017]). This fact suggests that training longer may have potential benefits on improving the robustness of the model. In our experiments, we also observe noticeable improvements of $L^2$-robustness on both the training set and test set (see Section 7.2).

2 Related Work

Implicit Bias in Training Linear Classifiers. For logistic regression (with linear predictor) on linearly separable data, [Soudry et al. 2018a,b] show that full-batch gradient descent converges in the direction of the max $L^2$-margin solution of the corresponding hard-margin Support Vector Machine (SVM). They also prove that the loss decreases at the rate of $O(1/t)$, the weight norm grows as $O(\log t)$, and the weight direction converges as $O(\log \log t/ \log t)$ (or $O(1/ \log t)$ if the dataset is non-degenerate). Subsequent works extend this result in several ways: [Nacson et al. 2018b] extends the results to the case of stochastic gradient descent, where the data is sampled in mini-batches without replacement; [Nacson et al. 2018a] considers other losses with poly-exponential tails; [Ji and Telgarsky 2018] characterizes the convergence
of weight direction for general datasets without assuming separability; [Gunasekar et al., 2018a] characterizes the convergence of weight direction for other optimization methods.

The results in [Soudry et al., 2018b,a] can also be generalized to deep linear networks. [Ji and Telgarsky, 2019] shows that the product of weights in a deep linear network with strictly decreasing loss converges in the direction of the max $L^2$-margin solution. [Gunasekar et al., 2018b] shows more general results for gradient descent on linear fully-connected and convolutional networks under various assumptions on the convergence of the loss and the gradient direction.

**Implicit Bias in Training Nonlinear Classifiers.** [Soudry et al., 2018a] analyze the case where there is only one trainable layer of a ReLU network. Our work is closely related to a very recent independent work by [Nacson et al., 2019], which we discuss in details below.

**Comparison with [Nacson et al., 2019].** [Nacson et al., 2019] analyzes gradient descent for smooth homogeneous models and proves the convergence of parameter direction to a KKT point of the aforementioned max-margin problem. Compared with their work, our work adopt much weaker assumptions: (1) They assume the training loss converges to 0, but in our work we only require that the training loss is lower than a small threshold value at some time $t_0$ (and we prove the exact convergence rate of the loss after $t_0$); (2) They assume the convergence of parameter direction, while we prove that KKT conditions hold for all limit points of $\{\theta(t)/\|\theta(t)\|_2 : t > 0\}$, without requiring any convergence assumption; (3) They assume the convergence of the direction of losses (the direction of the vector whose entries are loss values on every data point) and Linear Independence Constraint Qualification (LICQ) for the max-margin problem, while we do not need such assumptions. Besides the above differences in assumptions, we also prove the monotonicity of the normalized margin and provide tight convergence rate for training loss. We believe both results are interesting in their own right.

Another technical difference is that their work analyzes discrete gradient descent on smooth homogeneous models (which fails to capture neural networks with ReLU activation). In our work, we analyze gradient flow on homogeneous models which could be non-smooth. We note that if we assume smoothness, it is not hard to adapt our analysis to discrete gradient descent.

**Other Works on Implicit Bias.** [Wilson et al., 2017], [Ali et al., 2018], [Gunasekar et al., 2018a] show that for the linear least-square problem gradient-based methods converge to the unique global minimum that is closest to the initialization in $L^2$ distance. [Du et al., 2019], [Jacot et al., 2018], [Lee et al., 2019], [Arora et al., 2019b] suggest that over-parameterized neural networks of sufficient width (or infinite width) evolve as linear models with Neural Tangent Kernel (NTK) and converges to a global minimum near the initial point. Other related works include [Ma et al., 2017], [Gidel et al., 2019], [Arora et al., 2019a], [Suggala et al., 2018], [Blanc et al., 2019], [Banburski et al., 2019], [Neyshabur et al., 2014, 2015].

## 3 Preliminaries

For any $N \in \mathbb{N}$, let $\{N\} = \{1, \ldots, N\}$. We use $\|v\|_p$ to denote the $L^p$-norm of a vector $v$. The default base of log is $e$.

A set-valued map $f : A \rightarrow B$ is a mapping from a set $A$ to $2^B$, where $2^B$ is the set consisting of all the subsets of $B$. For $x \in \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^d$, define $\text{dist}(x, Q) := \inf_{y \in Q} \|x - y\|_2$.

### 3.1 Deep Homogeneous Neural Networks

For $k > 0$, a function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is $k$-homogeneous if $F(\alpha x) = \alpha^k F(x)$ for all $x \in \mathbb{R}^d$ and $\alpha > 0$. For a neural network $\Phi$ parameterized by $\theta \in \mathbb{R}^d$, we use $\Phi(\theta; x) \in \mathbb{R}$ to denote the output of $\Phi$ on the input $x \in \mathbb{R}^d$. We say that a neural network $\Phi$ is $k$-homogeneous (with respect to its parameters) if $\Phi(\theta; x)$ is $k$-homogeneous with respect to $\theta$ for any $x \in \mathbb{R}^d$.

We mainly focus on binary classification in this paper. A dataset is denoted by $\mathcal{D} = \{(x_n, y_n) : n \in [N]\}$, where $x_n \in \mathbb{R}^d$ stands for a data input and $y_n \in \{\pm 1\}$ stands for the corresponding label. For a loss function

---

1Their paper was uploaded on ArXiv on May 19, 2019.
\( \ell : \mathbb{R} \to \mathbb{R}_{\geq 0}, \) we define the training loss of \( \Phi \) on the dataset \( \mathcal{D} \) to be \( \mathcal{L}(\theta) := \sum_{n=1}^{N} \ell(y_n \Phi(\theta; x_n)). \)

The margin for a single data point \( (x_n, y_n) \) is defined to be \( q_n(\theta) := y_n \Phi(\theta; x_n), \) and the margin for the entire dataset is defined to be \( q_{\min}(\theta) := \min_{n \in [N]} q_n(\theta). \) Let \( S^{d-1} = \{ \theta \in \mathbb{R}^d : \| \theta \|_2 = 1 \} \) be the set of \( L^2 \)-normalized parameters.

Let \( \rho := \| \theta \|_2 \) and \( \theta := \theta / \rho \in S^{d-1} \) be the length and direction of \( \theta \). For an \( L \)-homogeneous neural network, only \( \hat{\theta} \) is important for making predictions and \( q_{\min}(\epsilon \theta) = e^C q_{\min}(\theta) \) for all \( C > 0 \). So we define the normalized margin to be \( \hat{\gamma}(\theta) := q_{\min}(\theta) / \rho^L. \)

### 3.2 Training by Gradient Flow

A function \( f : X \to \mathbb{R} \) is locally Lipschitz if for every \( x \in X \) there exists a neighborhood \( U \) of \( x \) such that the restriction of \( f \) on \( U \) is Lipschitz continuous. For a locally Lipschitz function \( f : X \to \mathbb{R} \) with \( X \subseteq \mathbb{R}^d \), we define Clarke’s subdifferential \([\text{Clarke}, 1975] \text{ Clarke et al., 2008 Davis et al., 2019}\) for any \( x \in X \) as \( \partial f(x) := \text{conv} \{ \lim_{k \to \infty} \nabla f(x_k) : x_k \to x; f \text{ is differentiable at } x_k \}. \) Clarke’s subdifferential is a generalization of gradient in the sense that if \( f \) is continuously differentiable at \( x \), then \( \partial f(x) \) reduces to the singleton \( \{ \nabla f(x) \} \), where \( \nabla f(x) \) is the gradient of \( f \) at \( x \) in the usual sense. For brevity, we say that a function \( z : I \to \mathbb{R}^d \) on the interval \( I \) is an arc if \( z \) is absolutely continuous for any closed sub-interval of \( I \). If \( z \) is differentiable at \( t \), then we use \( z'(t) \) or \( \frac{dz}{dt}(t) \) to stand for the usual derivative. Following the terminology in \([\text{Davis et al., 2019}] \), we say that a locally Lipschitz function \( f : \mathbb{R}^d \to \mathbb{R} \) admits a chain rule if for any arc \( z : \mathbb{R}_{\geq 0} \to \mathbb{R}^d, \forall h \in \partial f(z(t)) \) : \( (f \circ z)'(t) = (h, z'(t)) \) holds for a.e. \( t > 0 \) (see also Appendix C).

We focus on the case of training by gradient flow. We can view the trajectory of parameter \( \theta \) during training as an arc \( \theta : \mathbb{R}_{\geq 0} \to \mathbb{R}^d, t \mapsto \theta(t) \) which satisfies the differential inclusion

\[
\frac{d\theta(t)}{dt} \in -\partial \mathcal{L}(\theta(t)) \quad \text{for a.e. } t \geq 0,
\]

where \( \partial \mathcal{L} : \mathbb{R}^d \rightrightarrows \mathbb{R}^d \) stands for Clarke’s subdifferential.

For convenience, we also view the functions of \( \theta \), including \( \mathcal{L}(\theta), q_n(\theta), q_{\min}(\theta) \), as functions of \( t \). So we can write \( \mathcal{L}(t) := \mathcal{L}(\theta(t)), q_n(t) := q_n(\theta(t)), q_{\min}(t) := q_{\min}(\theta(t)). \)

### 4 Homogeneous Model with Exponential Loss

In this section, we state our results for homogeneous models with exponential loss \( \ell(q) := e^{-q} \), for simplicity of presentation. Those results are corollaries of the theorems for more general losses (see Appendix A and B).

#### 4.1 Setting

In this setting, we assume the following:

(A1). (Regularity). For any fixed \( x, \Phi(\cdot; x) \) is locally Lipschitz and admits a chain rule;

(A2). (Homogeneity). For any fixed \( x, \Phi(\cdot; x) \) is \( L \)-homogeneous, i.e., \( \forall \alpha > 0 : \Phi(\alpha \theta; x) = \alpha^L \Phi(\theta; x); \)

(A3). (Exponential Loss). \( \ell(q) = e^{-q}; \)

(A4). (Separability). There exists a time \( t_0 \) such that \( \mathcal{L}(t_0) < 1. \)

(A1) is a regularity assumption about the network output. As shown in \([\text{Davis et al., 2019}] \), the output of almost every neural network admits a chain rule (as long as the neural network is composed of definable pieces in an o-minimal structure, e.g., ReLU, sigmoid, LeakyReLU).

(A2), (A3), (A4) correspond to the three conditions introduced in Section 1. Note that (A4) is an assumption about the separability since \( \mathcal{L}(t_0) < 1 \) ensures that \( q_n(t_0) > 0 \) for all \( n \in [N] \) by the properties of exponential loss. Here 1 is indeed the largest threshold such that this argument can work.
4.2 Smoothed Normalized Margin

Recall that the normalized margin is \( \tilde{\gamma}(\theta) := q_{\text{min}}(\theta)/\rho \). Since it is difficult to analyze \( \tilde{\gamma}(\theta) \) directly, we define the smoothed version of \( \tilde{\gamma}(\theta) \) as follows.

**Definition 4.1.** For \( \theta \in \mathbb{R}^d \), the smoothed normalized margin \( \tilde{\gamma}(\theta) \) of \( \theta \) is defined as

\[
\tilde{\gamma}(\theta) := \rho^{-L} \log \frac{1}{L}.
\]

Now we explain why \( \tilde{\gamma} \) is a good approximation of the normalized margin \( \gamma \). Observe that

\[
\log \frac{1}{L} = - \log \left( \sum_{n=1}^{N} e^{-q_n} \right) = -\text{LSE}(q_1, \ldots, q_N),
\]

where \( \text{LSE}(x_1, \ldots, x_N) = \log(\exp(x_1) + \cdots + \exp(x_N)) \) is the LogSumExp function. Note that \( x_{\text{max}} = \log(\exp(x_{\text{max}})) \leq \text{LSE}(x_1, \ldots, x_N) \leq \log(N \exp(x_{\text{max}})) = x_{\text{max}} + \log N \). Combining this with (1), we have

\[
\tilde{\gamma} - \rho^{-L} \log N \leq \gamma \leq \tilde{\gamma}.
\]

When \( \rho \to +\infty \), \( \gamma \) and \( \tilde{\gamma} \) converge to the same value since \( |\gamma - \tilde{\gamma}| \leq \rho^{-L} \log N \to 0 \).

4.3 Main Theorems

Now we state our first theorem, which says that the smoothed normalized margin \( \gamma \) is increasing over time. As \( \gamma \) and \( \tilde{\gamma} \) stay close to each other when \( \rho \) is large, this theorem also suggests that the normalized margin \( \gamma \) is increasing over time.

**Theorem 4.2 (Corollary of Theorem A.1).** The following are true about the margins:

1. For a.e. \( t > t_0 \), \( \frac{d\gamma}{dt} \geq 0 \);
2. For a.e. \( t > t_0 \), either \( \frac{d\gamma}{dt} > 0 \) or \( \frac{d\tilde{\gamma}}{dt} = 0 \);
3. \( L \to 0 \) and \( \rho \to +\infty \) as \( t \to +\infty \); therefore, \( |\gamma(t) - \tilde{\gamma}(t)| \to 0 \).

It is easy to show that both \( \gamma(\theta) \) and \( \tilde{\gamma}(\theta) \) can be upper-bounded by a constant using the continuity of \( q_n(\theta) \) on \( S^{d-1} = \{ \theta \in \mathbb{R}^d : ||\theta||_2 = 1 \} \) and the inequality (2). Further, by Theorem 4.2 and the monotone convergence theorem, we know that \( \lim_{t \to +\infty} \gamma(t) \) and \( \lim_{t \to +\infty} \tilde{\gamma}(t) \) exist and equal to the same value.

To understand the implicit regularization effect, a natural question is: what optimality property does the limit of normalized margin have? To this end, we consider the following natural constrained optimization problem (P):

\[
\min \frac{1}{2} ||\theta||_2^2 \quad \text{s.t.} \quad q_n(\theta) \geq 1 \quad \forall n \in [N]
\]

For a linear model where \( q_n(\theta) := y_n \theta^T x_n \), we can see that (P) reduces to the optimization problem for the classic hard \( L^2 \)-margin Support Vector Machine (SVM). Hence, (P) can be seen as a generalization of SVM to deep neural networks. Indeed, minimizing (P) over its feasible region is equivalent to maximizing the normalized margin over all possible directions. The proof is simple: due to homogeneity, \( \theta \) obtains a 100% training accuracy iff \( \theta / q_{\text{min}}(\theta)^{1/L} \) is a feasible point of (P). A simple calculation can show that the objective value at \( \theta / q_{\text{min}}(\theta)^{1/L} \) is \( \frac{1}{2} \gamma^*(\theta)^{-2/L} \). Therefore, the maximum normalized margin is \( \gamma^* \), iff the minimum objective of (P) is \( \frac{1}{2} \gamma^*^{-2/L} \).

Since (P) is a highly nonconvex and nonsmooth problem, we cannot hope for proving a very strong convergence result like \( \tilde{\gamma}(t) \) converges to the globally optimal normalized margin \( \gamma^* = \min_{\theta} \gamma(\theta) \). Indeed, as gradient descent is a first-order optimization method, the best we can hope for is perhaps to show the first-order optimality of \( \lim_{t \to +\infty} \tilde{\gamma}(t) \).

In the following theorem, we show that \( \lim_{t \to +\infty} \tilde{\gamma}(t) \) is indeed first-order optimal:

**Theorem 4.3 (Corollary of Theorem C.4).** For any limit point \( \theta \) of \( \{ \tilde{\theta}(t) : t \geq 0 \} \), \( \tilde{\theta} / q_{\text{min}}(\theta)^{1/L} \) is a KKT point of (P).

Here a feasible point of (P) is called a KKT point if it satisfies all the Karush-Kuhn-Tucker conditions. Since (P) satisfies MFCQ (See Lemma C.3), Karush-Kuhn-Tucker conditions are first-order necessary conditions for global optimality.
4.4 Proof Sketch

Lemma 4.4 below is the key lemma in our proof. It decomposes the growth of the smoothed normalized margin into the ratio of two quantities related to the radial and tangential velocity components of $\theta$ respectively. We will give a proof sketch for this later in this section. We believe that this lemma is of independent interest.

**Lemma 4.4** (Corollary of Lemma A.2). For a.e. $t > t_0$,

$$\frac{d}{dt} \log \rho > 0 \quad \text{and} \quad \frac{d}{dt} \log \gamma \geq L \left( \frac{d}{dt} \log \rho \right)^{-1} \left\| \frac{d\theta}{dt} \right\|_2^2.$$

Using Lemma 4.4 the first two claims in Theorem 4.2 can be directly proved. For the third claim, we make use the monotonicity of the margin to lower bound the gradient, and then show $\mathcal{L} \to 0$ and $\rho \to +\infty$. Combining this with (2) proves the third claim. We defer the detailed proof to Appendix A.

To show Theorem 4.3, we first change the time measure to $\log \rho$, i.e., now we see $t$ as a function of $\log \rho$. So the second inequality in Lemma 4.4 can be rewritten as $\frac{d\log \gamma}{d\log \rho} \geq L \left\| \frac{d\theta}{d\log \rho} \right\|_2^2$. Integrating on both sides and noting that $\gamma$ is upper-bounded, we know that there must be many instant $\log \rho$ such that $\left\| \frac{d\theta}{d\log \rho} \right\|_2^2$ is small. By analyzing the landscape of training loss, we show that these points are “approximate” KKT points. Then we show that every convergent sub-sequence of $\{\hat{\theta}(t) : t \geq 0\}$ can be modified to be a sequence of “approximate” KKT points which converges to the same limit. Then we conclude the proof by applying a theorem from [Dutta et al., 2013] to show that the limit of this convergent sequence of “approximate” KKT points is a KKT point. We defer the detailed proof to Appendix C.

Now we give a proof sketch for Lemma 4.4 in which we derive the formula of $\hat{\gamma}$ step by step.

**Proof Sketch of Lemma 4.4.** For ease of presentation, we ignore the regularity issues of taking derivatives in this proof sketch. We start from the equation $\frac{d\mathcal{L}}{dt} = -\left\langle \partial \mathcal{L}(\theta(t)) \left( \frac{d\theta}{dt} \right) \right\rangle = -\left\| \frac{d\theta}{dt} \right\|_2^2$ which follows from the chain rule (see also Lemma G.3). Then we note that $\frac{d\theta}{dt}$ can be decomposed into two parts: the radial component $v := \hat{\theta}^T \frac{d\theta}{dt}$ and the tangent component $u := (I - \hat{\theta}^T) \frac{d\theta}{dt}$.

The radial component is easier to analyze. By the chain rule, $\hat{\theta}^T \frac{d\theta}{dt} = \frac{1}{\rho} \left\langle \theta, \frac{d\theta}{dt} \right\rangle = \frac{1}{\rho} \cdot \frac{1}{2} \frac{d\rho^2}{dt}$. For $\frac{1}{2} \frac{d\rho^2}{dt}$, we have

$$\frac{1}{2} \frac{d\rho^2}{dt} = \left\langle \theta, \frac{d\theta}{dt} \right\rangle = \left\langle \sum_{n=1}^{N} e^{-q_n} \partial q_n \theta, \theta \right\rangle = L \sum_{n=1}^{N} e^{-q_n} q_n,$$

where the last equality is due to $\left\langle \partial q_n \theta, \theta \right\rangle = L q_n$ by homogeneity of $q_n$. This equation is sometimes called Euler’s Theorem for Homogeneous Functions (see Theorem A.4). For differentiable functions, it can be easily proved by taking the derivative over $c$ on both sides of $q_n(c \theta) = c^T q_n(\theta)$ and letting $c = 1$.

With (3), we can lower bound $\frac{1}{2} \frac{d\rho^2}{dt}$ by

$$\frac{1}{2} \frac{d\rho^2}{dt} = L \sum_{n=1}^{N} e^{-q_n} q_n \geq L \sum_{n=1}^{N} e^{-q_n} \min q_n \geq L \cdot \mathcal{L} \log \frac{1}{\mathcal{L}},$$

where the last inequality uses the fact that $e^{-q_{\min}} \leq \mathcal{L}$. (4) also implies that $\frac{1}{2} \frac{d\rho^2}{dt} > 0$ for $t > t_0$ since $\mathcal{L}(t_0) < 1$ and $\mathcal{L}$ is non-increasing. As $\frac{d}{dt} \log \rho = \frac{1}{2} \frac{d\rho^2}{dt}$, this also proves the first inequality of Lemma 4.4.

Now, we have $\left\| v \right\|_2^2 = \frac{1}{\rho^2} \left( \frac{1}{2} \frac{d\rho^2}{dt} \right)^2$ on the one hand; on the other hand, by the chain rule we have

$$\frac{d\hat{\theta}}{dt} = \frac{\hat{\theta}}{\rho^2} (\rho \frac{d\theta}{dt} - \frac{d\rho}{dt} \theta) = \frac{1}{\rho^2} (\rho \frac{d\theta}{dt} - (\hat{\theta}^T \frac{d\theta}{dt}) \theta) = \frac{u}{\rho}.$$
So we have
\[- \frac{dL}{dt} = \left\| \frac{d\theta}{dt} \right\|^2_2 = \|v\|^2_2 + \|u\|^2_2 = \frac{1}{\rho_2^2} \left( \frac{1}{2} \frac{d\rho_2}{dt} \right)^2 + \rho_2^2 \left\| \frac{d\theta}{dt} \right\|^2_2\]

Dividing \( \frac{1}{2} \frac{d\rho_2}{dt} \) on both sides of \( - \frac{dL}{dt} = \frac{1}{\rho_2^2} \left( \frac{1}{2} \frac{d\rho_2}{dt} \right)^2 + \rho_2^2 \left\| \frac{d\theta}{dt} \right\|^2_2 \), we have
\[- \frac{dL}{dt} \left( \frac{1}{2} \frac{d\rho_2}{dt} \right)^{-1} = \frac{d}{dt} \log \rho + \left( \frac{d}{dt} \log \rho \right)^{-1} \left\| \frac{d\theta}{dt} \right\|^2_2.

By \( - \frac{dL}{dt} \geq 0 \) and (4), we further have
\[- \frac{dL}{dt} \left( L \cdot \mathcal{L} \log \frac{1}{\mathcal{L}} \right)^{-1} \geq \frac{d}{dt} \log \rho + \left( \frac{d}{dt} \log \rho \right)^{-1} \left\| \frac{d\theta}{dt} \right\|^2_2,

and so \( \frac{d}{dt} \log \log \frac{1}{\mathcal{L}} - L \frac{d}{dt} \log \rho \geq L \left( \frac{d}{dt} \log \rho \right)^{-1} \left\| \frac{d\theta}{dt} \right\|^2_2 \), where the LHS is exactly \( \frac{d}{dt} \log \gamma \). \( \square \)

## 5 Homogeneous Model with General Loss

In this section, we generalize the results in Section 4 to a much broader class of binary classification loss. A major consequence of this generalization is that the logistic loss, one of the most popular loss functions, \( \ell(q) = \log(1 + e^{-q}) \) is included. The function class also includes other losses with exponential tail, e.g., \( \ell(q) = e^{-q^2}, \ell(q) = \log(1 + e^{-q^3}) \).

### 5.1 Setting

Now we state our assumptions in this setting. The first two assumptions are the same as those in Section 4. For (A3), (A4), we replace them with two new assumptions (B3), (A4):

(A1). (Regularity). For any fixed \( x, \Phi(\cdot; x) \) is locally Lipschitz and admits a chain rule;

(A2). (Homogeneity). For any fixed \( x, \Phi(\cdot; x) \) is \( L \)-homogeneous, i.e., \( \forall \alpha > 0 : \Phi(\alpha \theta; x) = \alpha^L \Phi(\theta; x) \);

(B3). The loss function \( \ell(q) \) can be expressed as \( \ell(q) = e^{-f(q)} \) such that

(B3.1). \( f : \mathbb{R} \to \mathbb{R} \) is \( C^1 \)-smooth.

(B3.2). \( f'(q) > 0 \) for all \( q \in \mathbb{R} \).

(B3.3). There exists \( b_f \geq 0 \) such that \( f'(q)q \) is non-decreasing for \( q \in (b_f, +\infty) \), and \( f'(q)q \to +\infty \) as \( q \to +\infty \).

(B3.4). Let \( g : [f(b_f), +\infty) \to [b_f, +\infty) \) be the inverse function of \( f \) on the domain \( [b_f, +\infty) \). There exists \( b_g \geq \max\{2f(b_f), f(2b_f)\}, K > 0 \) such that \( g'(x) \leq Kg'(\theta x) \) and \( f'(y) \leq K f'(\theta y) \) for all \( x \in (b_g, +\infty), y \in (g(b_g), +\infty) \) and \( \theta \in [1/2, 1] \).

(B4). (Separability). There exists a time \( t_0 \) such that \( \mathcal{L}(t_0) < e^{-f(b_f)} = \ell(b_f) \).

(A1) and (A2) remain unchanged. (B3) is satisfied by exponential loss \( \ell(q) = e^{-q} \) (with \( f(q) = q \)) and logistic loss \( \ell(q) = \log(1 + e^{-q}) \) (with \( f(q) = -\log \log(1 + e^{-q}) \)). (B4) are essentially the same as (A4) but (B4) uses a threshold value that depends on the loss function. Assuming (B3), it is easy to see that (B4) ensures the separability of data since \( \ell(q_n) < e^{-f(b_f)} \) implies \( q_n > b_f \geq 0 \). For logistic loss, we can set \( b_f = 0 \) (which will be proved in Lemma 5.1 later), so the corresponding threshold value in (B4) is \( \ell(0) = \log 2 \).

Now we discuss each of the assumptions in (B3). (B3.1) is a natural assumption on smoothness. (B3.2) requires \( \ell(\cdot) \) to be monotone decreasing, which is also natural since \( \ell(\cdot) \) is used for binary classification. The rest of two assumptions in
(B3) characterize the properties of $\ell'(q)$ when $q$ is large enough. (B3.3) is an assumption that appears naturally from the proof. For exponential loss, $f'(q)q = q$ is always non-decreasing, so we can set $b_f = 0$. In (B3.4), the inverse function $g$ is defined. It is guaranteed by (B3.1) and (B3.2) that $g$ always exists and is also $C^1$-smooth. Though (B3.4) looks very complicated, it essentially says that $f'(\Theta(q)) = \Theta(f'(q))$, $g'(\Theta(q)) = \Theta(g'(q))$ as $q \to \infty$. (B3.4) is indeed a technical assumption that enables us to asymptotically compare the loss or the length of gradient at different data points. It is possible to base our results on weaker assumptions than (B3.4), but we use (B3.4) for simplicity since it has already been satisfied by many loss functions such as the aforementioned examples. For exponential loss, $f(q) = g(q) = q$, so $f''(q) = g'(q) = 1$, which trivially satisfies (B3.4). We prove in the following lemma that the logistic loss satisfies (B3).

**Lemma 5.1.** The logistic loss $\ell(q) = \log(1 + e^{-q})$ satisfies (B3) with $b_f = 0$, $f'(q) = \Theta(1)$, $g'(x) = \Theta(1)$.

**Proof.** A simple calculation shows that $f(q) = -\log(1 + e^{-q})$ and $g(x) = -\log(\exp(-x)) - 1$. (B3.1) is trivial. $f'(q) = \frac{e^{-q}}{(1 + e^{-q}) \log(1 + e^{-q})} > 0$, so (B3.2) is satisfied. For (B3.3), note that $f'(q)q = \frac{q}{(1 + e^q) \log(1 + e^{-q})}$. The denominator is a decreasing function since

$$
\frac{d}{dq} \left( (1 + e^q) \log(1 + e^{-q}) \right) = e^q \log(1 + e^{-q}) - 1 < e^q \cdot e^{-q} - 1 = 0.
$$

Thus, $f'(q)q$ is a strictly increasing function on $\mathbb{R}$. As $b_f$ is required to be non-negative, we set $b_f = 0$. For proving that $f'(q)q \to +\infty$ and (B4), we only need to notice that $f'(q) \sim \frac{e^{-q}}{1 + e^{-q}} = 1$ and $g'(x) = 1/f'(g(x)) \sim 1$. \hfill \Box

### 5.2 Smoothed Normalized Margin

For a loss function $\ell(\cdot)$ satisfying (B3), it is easy to see from (B3.2) that its inverse function $\ell^{-1}(\cdot)$ must exist. For this kind of loss functions, we define the smoothed normalized margin as follows:

**Definition 5.2.** For a loss function $\ell(\cdot)$ satisfying (B3), the smoothed normalized margin $\tilde{\gamma}(\theta)$ of $\theta$ is defined as

$$
\tilde{\gamma}(\theta) := \ell^{-1}(\mathcal{L})/\psi^L,
$$

where $\ell^{-1}(\cdot)$ is the inverse function of $\ell(\cdot)$.

**Remark 5.3.** For logistic loss $\ell(q) = \log(1 + e^{-q})$, $\tilde{\gamma}(\theta) = \rho^{-L} \log \frac{1}{\exp(\mathcal{L}) - 1}$; for exponential loss $\ell(q) = e^{-q}$, $\tilde{\gamma}(\theta) = \rho^{-L} \log \mathcal{L}$, which is the same as in Definition 5.2.

Similar to the exponential loss case, we can show that $\tilde{\gamma}$ defined in Definition 5.2 is a good approximation for $\tilde{\gamma}$. Further, Theorem 4.2 and Theorem 4.3 continues to hold in this setting. We defer the proofs for these two theorems to Appendix A and B.

Now we explain why $\tilde{\gamma}$ is a good approximation for $\tilde{\gamma}$ using a similar argument as in Section 4.2. By definition of $g(\cdot)$, the smoothed normalized margin $\tilde{\gamma}(\theta)$ can also be written as $g(\log \frac{1}{\mathcal{L}})/\psi^L$ for $\mathcal{L} < b_f$. Then we have

$$
\tilde{\gamma}(\theta) \psi^L = g(\log \frac{1}{\mathcal{L}}) = g \left( \log \left( \frac{1}{\sum_{n=1}^{N} \frac{1}{e^{-f(q_n)}}} \right) \right) = g \left( -\text{LSE}(\{f(q_1), \ldots, f(q_N)\}) \right),
$$

where $\text{LSE}(x_1, \ldots, x_N) = \log(\exp(x_1) + \cdots + \exp(x_N))$ is the LogSumExp function. Again, using the inequality $x_{\max} \leq \text{LSE}(x_1, \ldots, x_N) \leq x_{\max} + \log N$, we can do the following approximation:

$$
\tilde{\gamma}(\theta) \psi^L = g \left( -\text{LSE}(\{-f(q_1), \ldots, -f(q_N)\}) \right) \approx g(-\min\{-f(q_1), \ldots, -f(q_N)\}) = g(f(q_{\min})) = q_{\min},
$$

which implies that $\tilde{\gamma} \approx q_{\min}/\psi^L = \tilde{\gamma}$. Note that (B3.3) is crucial to make the approximation reasonable. Using a more rigorous analysis, we can show the following lemma.

**Lemma 5.4.** We have the following properties about the margin:

(a) $f(q_{\min}) - \log N \leq \log \frac{1}{\mathcal{L}} \leq f(q_{\min})$. 

(b) If \( \log \frac{1}{b} > f(b_f) \), then there exists \( \xi \in (f(q_{\text{min}}) - \log N, f(q_{\text{min}})) \cap (b_f, +\infty) \) such that

\[
\tilde{\gamma} - \frac{g'(\xi) \log N}{\rho^L} \leq \gamma \leq \tilde{\gamma}.
\]

(c) Let \( \{\theta_m \in \mathbb{R}^d : m \in \mathbb{N}\} \) be a sequence of parameters. If \( \mathcal{L}(\theta_m) \to 0 \), then \( |\tilde{\gamma}(\theta_m) - \tilde{\gamma}(\theta_m)| \to 0 \).

Proof for Lemma 5.4 (a) can be easily deduced from \( e^{-f(q_{\text{min}})} \leq \mathcal{L} \leq N e^{-f(q_{\text{min}})} \). Combining (a) and the monotonicity of \( g(\cdot) \), we further have \( g(s) \leq g(\log \frac{1}{b}) \leq q_{\text{min}} \) for \( s := \max\{f(b_f), f(q_{\text{min}}) - \log N\} \). By the mean value Theorem, there exists \( \xi \in (s, f(q_{\text{min}})) \) such that \( g(s) = g(f(q_{\text{min}})) - g'(\xi)(f(q_{\text{min}}) - s) \geq q_{\text{min}} - g'(\xi) \log N \). Dividing \( \rho^L \) on each side of \( q_{\text{min}} - g'(\xi) \log N \leq g(\log \frac{1}{b}) \leq q_{\text{min}} \) proves (b).

Now we prove (c). Without loss of generality, we assume \( \log \frac{1}{\mathcal{L}(\theta_m)} > f(b_f) \) for all \( \theta_m \). It follows from (b) that for every \( \theta_m \) there exists \( \xi_m \in (f(q_{\text{min}}(\theta_m)) - \log N, f(q_{\text{min}}(\theta_m))) \cap (b_f, +\infty) \) such that

\[
\tilde{\gamma}(\theta_m) - \frac{g'(\xi_m) \log N}{\rho(\theta_m)^L} \leq \gamma(\theta_m) \leq \tilde{\gamma}(\theta_m).
\]

Note that \( \xi_m \geq f(q_{\text{min}}(\theta_m)) - \log N \geq \log \frac{1}{\mathcal{L}(\theta_m)} - \log N \to +\infty \). So \( \frac{g(\xi_m)}{g'(\xi_m)} = \frac{g'(\xi_m)g(\xi_m)}{g'(\xi_m)} \to +\infty \) by (B3.3).

Also note that there exists a constant \( B_0 \) such that \( \tilde{\gamma}(\theta_m) \leq B_0 \) for all \( m \) since \( \tilde{\gamma} \) is continuous on the unit sphere \( S^{d-1} \). So we have

\[
\frac{g'(\xi_m)}{\rho(\theta_m)^L} = \frac{g'(\xi_m)}{g(\xi_m)} \frac{g(\xi_m)}{\rho(\theta_m)^L} \leq \frac{g'(\xi_m)}{g(\xi_m)} \frac{\min(\theta_m)}{\rho(\theta_m)^L} = \frac{g'(\xi_m)}{g(\xi_m)} \frac{\gamma(\theta_m)}{\min(\theta_m)} : B \to 0,
\]

where the first inequality follows since \( \xi_m \leq f(q_{\text{min}}(\theta_m)) \). Together with (6), we have \( |\tilde{\gamma}(\theta_m) - \gamma(\theta_m)| \to 0 \).

6 Other Extensions

Cross-entropy Loss. While the main focus of this paper is binary classification, we also show that similar results hold for multi-class classification with cross-entropy loss. In multi-class classification, we define \( q_s \) to be the difference between the classification score for the true label and the maximum score for the other labels. And we define the margin \( q_{\text{min}} := \min_{n \in [N]} q_n \) and the normalized margin \( \tilde{\gamma} := q_{\text{min}}/\rho^L \) as before. In Appendix D, we define the smoothed normalized margin for cross-entropy loss to be the same as that for logistic loss (See Remark 5.3, and Lemma 4.4 continues to hold, and thus Theorem 4.2 and Theorem 4.3 still hold (but with a slightly different definition of (P)).

Multi-homogeneous Models. Some neural networks indeed possess a stronger property than homogeneity, which we call multi-homogeneity. For example, the output of a CNN (without bias terms) is 1-homogeneous with respect to the weights of each layer. In general, we say that a neural network \( \Phi(\theta; x) \) with \( \theta = (w_1, \ldots, w_m) \) is \((k_1, \ldots, k_m)\)-homogeneous if for any \( x \) and any \( c_1, \ldots, c_m > 0 \), we have \( \Phi(c_1 w_1, \ldots, c_m w_m; x) = \prod_{i=1}^m c_i^{k_i} \Phi(w_1, \ldots, w_m; x) \).

In the previous example, an \( L \)-layer CNN with layer weights \( \theta = (w_1, \ldots, w_L) \) is \((1, \ldots, 1)\)-homogeneous.

One can easily see that that \((k_1, \ldots, k_m)\)-homogeneity implies \( L \)-homogeneity, where \( L = \sum_{i=1}^m k_i \), so our previous analysis for homogeneous models still applies to multi-homogeneous models. But it would be better to define the normalized margin for multi-homogeneous model as

\[
\tilde{\gamma}(w_1, \ldots, w_m) := \min_{n \in [N]} \left( \frac{w_1}{\|w_1\|_2}, \ldots, \frac{w_m}{\|w_m\|_2} \right) = \frac{q_{\text{min}}}{\prod_{i=1}^m \|w_i\|_2^{k_i}}.
\]

(7)

In this case, the smoothed approximation of \( \tilde{\gamma} \) for general binary classification loss (under some conditions) can be similarly defined:

\[
\tilde{\gamma}(w_1, \ldots, w_m) := \frac{\ell^{-1}(\mathcal{L})}{\prod_{i=1}^m \|w_i\|_2^{k_i}},
\]

(8)

It can be shown that \( \tilde{\gamma} \) is also increasing during training when the loss is small enough (Appendix E). In the case of cross-entropy loss, we can also define \( \tilde{\gamma} \) by (9) while \( \ell(\cdot) \) is set to the logistic loss in the formula.
Convergence Rate. In Theorem 4.2, it is shown that $\mathcal{L} \to 0$ and $\rho \to \infty$. Indeed, a refined analysis can show that $\mathcal{L}(t) = \Theta\left(\frac{1}{(\log t)^{1/L}}\right)$ and $\rho(t) = \Theta\left((\log t)^{1/L}\right)$ for $L$-homogeneous neural network with exponential loss or logistic loss. We present this analysis in Appendix F.

7 Experiments

To validate our theoretical results, we conduct several experiments on MNIST with Tensorflow. We trained two models. The first one (called the CNN with bias) is a standard 4-layer CNN with exactly the same architecture as that used in MNIST Adversarial Examples Challenge. The layers of this model can be described as conv-32 with filter size $5 \times 5$, max-pool, conv-64 with filter size $3 \times 3$, max-pool, fc-1024, fc-10 in order. Notice that this model has bias terms in each layer, and thus does not satisfy homogeneity. To make its outputs homogeneous to its parameters, we also trained this model after removing all the bias terms except those in the first layer (the modified model is called the CNN without bias). Note that keeping the bias terms in the first layer prevents the model to be homogeneous in the input data while retains the homogeneity in parameters. We initialize all layer weights by He normal initializer [He et al., 2015] and all bias terms by zero. In training the models, we use SGD with batch size 100 without momentum. We normalize all the images to $[0, 1]^{32 \times 32}$ by dividing 255 for each pixel.

7.1 Evaluation for Normalized Margin

In the first part of our experiments, we evaluate the normalized margin every few epochs to see how it changes over time. From now on, we view the bias term in the first layer as a part of the weight in the first layer for convenience. Observe that the CNN without bias is multi-homogeneous in layer weights (See Section 6). So for the CNN without bias, we define the normalized margin $\bar{\gamma}$ as the margin divided by the product of the $L_2$-norm of all layer weights. Here we compute the $L_2$-norm of a layer weight parameter after flattening it into a one-dimensional vector. For the CNN with bias, we still compute the smoothed normalized margin in this way. When computing the $L_2$-norm of every layer weight, we simply ignore the bias terms if they are not in the first layer. For completeness, we include the plots for the normalized margin using the original definition in Appendix H.2.

SGD with Constant Learning Rate. We first train the CNNs using SGD with constant learning rate 0.01. After about 100 epochs, both CNNs have fitted the training set. After that, we can see that the normalized margins of both CNNs increase. However, the growth rate of the normalized margin is rather slow. See Figure 1 for more details.

SGD with Loss-based Learning Rate. Indeed, we can speed up the training by using a proper scheduling of learning rates for SGD. We propose a heuristic learning rate scheduling method, called the loss-based learning rate scheduling. The basic idea is to find the maximum possible learning rate at each epoch based on the current training loss (in a similar way as the line search method). See Appendix H.1 for the details. As shown in Figure 1, SGD with loss-based learning rate scheduling decreases the training loss exponentially faster than SGD with constant learning rate. Also, a rapid growth of normalized margin is observed for both CNNs. Note that with this scheduling the training loss can be as small as $10^{-800}$, which may lead to numerical issues. To address such issues, we applied some re-parameterization tricks and numerical tricks in our implementation. See Appendix H.3 for the details.

7.2 Evaluation for Robustness

Recently, robustness of deep learning has received considerable attention [Szegedy et al., 2013, Biggio et al., 2013, Athalye et al., 2018], since most state-of-the-arts deep neural networks are found to be very vulnerable against small but adversarial perturbations of the input points. In our experiments, we found that enlarging the normalized margin can improve the robustness. In particular, by simply training the neural network for a longer time with our loss-based learning rate, we observe noticeable improvements of $L_2$-robustness on both the training set and test set.

https://github.com/MadryLab/mnist_challenge
We first elaborate the relationship between the normalized margin and the robustness from a theoretical perspective. For a data point \( z = (x, y) \), we can define the robustness (with respect to some norm \( \| \cdot \| \)) of a neural network \( \Phi \) for \( z \) to be

\[
R_\theta(z) := \inf_{x' \in X} \{ \| x - x' \| : (x', y) \text{ is misclassified} \}.
\]

where \( X \) is the data domain (which is \([0, 1]^{32 \times 32}\) for MNIST). It is well-known that the normalized margin is a lower bound of \( L^2 \)-robustness for fully-connected networks (See, e.g., Theorem 4 in [Sokolic et al., 2017]). Indeed, a general relationship between those two quantities can be easily shown. Note that a data point \( z \) is correctly classified iff the margin for \( z \), denoted as \( q_\theta(z) \), is larger than 0. For homogeneous models, the margin \( q_\theta(z) \) and the normalized margin \( q_\theta(z) \) for \( x \) have the same sign. If \( q_\theta(\cdot) : \mathbb{R}^{d_x} \rightarrow \mathbb{R} \) is \( \beta \)-Lipschitz (with respect to some norm \( \| \cdot \| \)), then it is easy to see that \( R_\theta(z) \geq q_\theta(z)/\beta \). This suggests that improving the normalize margin on the training set can improve the robustness on the training set. Therefore, our theoretical analysis suggests that training longer can improve the robustness of the model on the training set.

This observation does match with our experiment results. In the experiments, we measure the \( L^2 \)-robustness of the CNN without bias for the first time its loss decreases below \( 10^{-10}, 10^{-15}, 10^{-20}, 10^{-120} \) (labelled as model-1 to model-4 respectively). We also measure the \( L^2 \)-robustness for the final model after training for 10000 epochs (labelled as model-5), whose training loss is about \( 10^{-882} \). The normalized margin of each model is monotone increasing with respect to the number of epochs, as shown in Table 1.

We use the standard method for evaluating \( L^2 \)-robustness in [Carlini and Wagner, 2017] and the source code from the
Figure 2: $L^2$-robustness of the models of CNNs without bias trained for different number of epochs (see Table 1 for the statistics of each model). Figures on the first row show the robust accuracy on the training set, and figures on the second row show that on the test set. On every row, the left figure and the right figure plot the same curves but they are in different scales. From model-1 to model-4, noticeable robust accuracy improvements can be observed. The improvement of model-5 upon model-4 is marginal or nonexistent for some $\epsilon$, but the improvement upon model-1 is always significant.

Table 1: Statistics of the CNN without bias after training for different number of epochs.

| model name | number of epochs | train loss | normalized margin | train acc | test acc |
|------------|------------------|------------|-------------------|-----------|---------|
| model-1    | 38               | $10^{-10.04}$ | $5.65 \times 10^{-5}$ | 100%      | 99.3%   |
| model-2    | 75               | $10^{-15.12}$ | $9.50 \times 10^{-5}$ | 100%      | 99.3%   |
| model-3    | 107              | $10^{-20.07}$ | $1.30 \times 10^{-4}$ | 100%      | 99.3%   |
| model-4    | 935              | $10^{-120.01}$ | $4.61 \times 10^{-4}$ | 100%      | 99.2%   |
| model-5    | 10000            | $10^{-881.51}$ | $1.18 \times 10^{-3}$ | 100%      | 99.1%   |

authors with default hyperparameter. We plot the robust accuracy (the percentage of data with robustness $> \epsilon$) for the training set in the figures on the first row of Figure 2. It can be seen from the figures that for small $\epsilon$ (e.g., $\epsilon < 0.3$), the relative order of robust accuracy is just the order of model-1 to model-5. For relatively large $\epsilon$ (e.g., $\epsilon > 0.3$), the improvement of model-5 upon model-2 to model-4 becomes marginal or nonexistent in certain intervals of $\epsilon$, but model-1 to model-4 still have an increasing order of robust accuracy and the improvement of model-5 upon model-1 is always significant. This shows that training longer can help to improve the $L^2$-robust accuracy on the training set.

We also evaluate the robustness on the test set, in which a misclassified test sample is considered to have robustness 0, and plot the robust accuracy in the figures on the second row of Figure 2. It can be seen from the figures that for small $\epsilon$ (e.g., $\epsilon < 0.2$), the curves of the robust accuracy of model-1 to model-5 are almost indistinguishable. However, for relatively large $\epsilon$ (e.g., $\epsilon > 0.2$), again, model-1 to model-4 have an increasing order of robust accuracy and the improvement of model-5 upon model-1 is always significant. This shows that training longer can also help to improve the $L^2$-robust accuracy on the test set.

We tried various different settings of hyperparameters for the evaluation method (including different learning rates, https://github.com/carlini/nn_robust_attacks
different binary search steps, etc.) and we observed that the shapes and relative positions of the curves in Figure 2 are stable across different hyperparameter settings.

8 Discussion and Future Directions

In this paper, we analyze the normalized margin of homogeneous neural networks under a minimal set of assumptions: Homogeneity, Separability, and No finite minima on the loss function. The main contribution of our work is to prove rigorously that for gradient flow, the normalized margin is increasing and converges to a KKT point of a natural max-margin problem. Our results leads to some natural further questions:

- Can we generalize our results to (stochastic) gradient descent? If the training loss is smooth, then it is not hard to discretize our analysis. However, modern neural networks are non-smooth due to the use of ReLU, and it remains unclear whether one can prove similar convergence results for SGD on non-smooth neural networks.

- Can we make more structural assumptions on the neural network to prove stronger results? In this work, we use a minimal set of assumptions to show that the convergent direction of parameters is a KKT point. A potential research direction is to identify more key properties of modern neural networks and show that the normalized margin at convergence is locally or globally optimal (in terms of optimizing (P)).

- Can we extend our results to neural networks with bias terms? In our experiments, the normalized margin of the CNN with bias also increases during training despite that its output is non-homogeneous. It would be interesting to provide a rigorous proof for this fact.

Acknowledgments

We thank Liwei Wang for helpful suggestions on the connection between margin and robustness. We thank Sanjeev Arora, Tianle Cai, Simon Du, Jason D. Lee, Zhiyuan Li, Tengyu Ma, Ruosong Wang for helpful discussions.

References

Alnur Ali, J Zico Kolter, and Ryan J Tibshirani. A continuous-time view of early stopping for least squares regression. arXiv preprint arXiv:1810.10082, 2018.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 242–252, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. arXiv preprint arXiv:1905.13655, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. arXiv preprint arXiv:1904.11955, 2019b.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Andrzej Banburski, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Bob Liang, Jack Hidary, and Tomaso Poggio. Theory III: Dynamics and generalization in deep networks. arXiv preprint arXiv:1903.04991, 2019.
Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. *arXiv preprint arXiv:1904.09080*, 2019.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, May 2017. doi: 10.1109/SP.2017.49.

Francis H. Clarke, Yuri S. Ledyaev, Ronald J. Stern, and Peter R. Wolenski. *Nonsmooth analysis and control theory*, volume 178. Springer Science & Business Media, 2008.

Frank H. Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

Frank H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990. doi: 10.1137/1.9781611971309.

Michel Coste. *An Introduction to O-minimal Geometry*. 2002.

Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, Jan 2019. ISSN 1615-3383. doi: 10.1007/s10208-018-09409-5.

Dmitriy Drusvyatskiy, Alexander D Ioffe, and Adrian S Lewis. Curves of descent. *SIAM Journal on Control and Optimization*, 53(1):114–138, 2015.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1675–1685, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Aarti Singh, and Barnabas Poczos. Gradient descent can take exponential time to escape saddle points. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1067–1077. Curran Associates, Inc., 2017.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 382–393. Curran Associates, Inc., 2018.

Joydeep Dutta, Kalyanmoy Deb, Rupesh Tulshyan, and Rammik Arora. Approximate KKT points and a proximity measure for termination. *Journal of Global Optimization*, 56(4):1463–1499, 2013.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in deep linear neural networks. *arXiv preprint arXiv:1904.13262*, 2019.

Giorgio Giorgi, Angelo Guarraggio, and Jörg Thierfelder. Chapter IV - Nonsmooth Optimization Problems. In *Mathematics of Optimization*, pages 359 – 457. Elsevier Science, Amsterdam, 2004. ISBN 978-0-444-50550-7.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018a. PMLR.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9482–9491. Curran Associates, Inc., 2018b.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In The IEEE International Conference on Computer Vision (ICCV), December 2015.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 8571–8580. Curran Associates, Inc., 2018.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. arXiv preprint arXiv:1803.07300, 2018.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In International Conference on Learning Representations, 2019.

Jaehoon Lee, Lecchao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. arXiv preprint arXiv:1902.06720, 2019.

Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution. arXiv preprint arXiv:1711.10467, 2017.

Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. arXiv preprint arXiv:1803.01905, 2018a.

Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. arXiv preprint arXiv:1806.01796, 2018b.

Mor Shpigel Nacson, Suriya Gunasekar, Jason Lee, Nathan Srebro, and Daniel Soudry. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 4683–4692, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. arXiv preprint arXiv:1412.6614, 2014.

Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-SGD: Path-normalized optimization in deep neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2422–2430. Curran Associates, Inc., 2015.

Saharon Rosset, Ji Zhu, and Trevor J. Hastie. Margin maximizing loss functions. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, Advances in Neural Information Processing Systems 16, pages 1237–1244. MIT Press, 2004.

Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel R. D. Rodrigues. Robust large margin deep neural networks. IEEE Trans. Signal Processing, 65(16):4265–4280, 2017. doi: 10.1109/TSP.2017.2708039.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. Journal of Machine Learning Research, 19, 2018a.

Daniel Soudry, Elad Hoffer, and Nathan Srebro. The implicit bias of gradient descent on separable data. In International Conference on Learning Representations, 2018b.

Arun Suggala, Adarsh Prasad, and Pradeep K Ravikumar. Connecting optimization and regularization paths. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 10608–10619. Curran Associates, Inc., 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In International Conference on Learning Representations, 2013.

Lou van den Dries and Chris Miller. Geometric categories and o-minimal structures. Duke Mathematical Journal, 84 (2):497–540, 1996.
Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. \textit{arXiv preprint arXiv:1810.05369}, 2018.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, \textit{Advances in Neural Information Processing Systems 30}, pages 4148–4158. Curran Associates, Inc., 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In \textit{International Conference on Learning Representations}, 2017.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In \textit{International Conference on Learning Representations}, 2019.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks, 2018.
A Margin Monotonicity for General Loss

In this section, we prove the monotonicity of the normalized margin for general loss. We assume (A1), (A2), (B3), (B4) as mentioned in Section 5. But (B3.4) is indeed not needed for showing the results in this section.

Theorem A.1. For \( \tilde{\gamma} \) defined in Definition 5.2, the following are true about the margins:

1. For a.e. \( t > t_0 \), \( \frac{d}{dt} \tilde{\gamma} \geq 0 \);
2. For a.e. \( t > t_0 \), either \( \frac{d}{dt} \tilde{\gamma} > 0 \) or \( \frac{d}{dt} \hat{\theta} = 0 \).
3. \( L \to 0 \) and \( \rho \to \infty \) as \( t \to +\infty \); therefore, \( |\bar{\gamma}(t) - \tilde{\gamma}(t)| \to 0 \).

To prove the first two propositions, we generalize our key lemma (Lemma 4.4) to general loss. The proof can be found in Appendix A.1.

Lemma A.2. For \( \tilde{\gamma} \) defined in Definition 5.2, the following holds for all \( t > t_0 \),

\[
\frac{d}{dt} \log \rho > 0 \quad \text{and} \quad \frac{d}{dt} \log \tilde{\gamma} \geq L \left( \frac{d}{dt} \log \rho \right) \left[ \frac{\| d\hat{\theta} / dt \|^2}{2} \right].
\] (9)

To prove the third proposition, we prove the following lemma to show that \( L \to 0 \) by giving an upper bound for \( L \). Since \( L \) can never be 0 for bounded \( \rho \), \( L \to 0 \) directly implies \( \rho \to +\infty \). The proof for this can be found in Appendix A.2.

For showing \( |\bar{\gamma} - \tilde{\gamma}| \to 0 \), we only need to apply (c) in Lemma 5.4 which shows this when \( L \to 0 \).

A.1 Key Lemma

Before proving Lemma A.2, we review two important properties of homogeneous functions. Note that these two properties are usually shown for smooth functions. By considering Clarke’s subdifferential, we can generalize it to locally Lipschitz functions that admit chain rules:

Theorem A.4. Let \( F : \mathbb{R}^d \to \mathbb{R} \) be a locally Lipschitz function that admits a chain rule. If \( F \) is \( k \)-homogeneous, then

(a) For all \( x \in \mathbb{R}^d \) and \( \alpha > 0 \),

\[
\partial^\alpha F(\alpha x) = \alpha^{k-1} \partial^\alpha F(x)
\]

That is, \( \partial^\alpha F(\alpha x) = \{ \alpha^{k-1} h : h \in \partial^\alpha F(x) \} \).

(b) (Euler’s Theorem for Homogeneous Functions). For all \( x \in \mathbb{R}^d \),

\[
\langle x, \partial^\alpha F(x) \rangle = k \cdot F(x)
\]

That is, \( \langle x, h \rangle = k \cdot F(x) \) for all \( h \in \partial^\alpha F(x) \).
Applying Theorem A.4 to homogeneous neural networks, we have the following corollary:

**Theorem A.6.**

Taking limits by Lemma 5.4, for convenience, we define

\[
\nabla \text{differetiability and gradient, }
\n\nabla \text{subdifferential, for proving (a), it is sufficient to show that}
\]

\[
\{ \lim_{k \to \infty} \nabla F(\alpha x_k) : x_k \to x, \alpha x_k \in D \} = \{ \alpha^{k-1} \lim_{k \to \infty} \nabla F(x_k) : x_k \to x, x_k \in D \}
\]

(10)

Fix \( x_k \in D \). Let \( U \) be a neighborhood of \( x_k \). By definition of homogeneity, for any \( h \in \mathbb{R}^d \) and any \( y \in U \setminus \{ x_k \} \),

\[
\frac{F(\alpha y) - F(\alpha x_k) - \langle \alpha y - \alpha x_k, \alpha^{k-1} h \rangle}{\| \alpha y - \alpha x_k \|_2} = \alpha^{k-1} \left[ \frac{F(y) - F(x_k) - (y - x_k, h)}{\| y - x_k \|_2} \right].
\]

Taking limits \( y \to x_k \) on both sides, we know that the LHS converges to 0 iff the RHS converges to 0. Then by definition of differentiability and gradient, \( F \) is differentiable at \( \alpha x_k \) iff it is differentiable at \( x_k \), and \( \nabla F(\alpha x_k) = \alpha^{k-1} h \) iff \( \nabla F(x_k) = h \). This proves (10).

To prove (b), we fix \( x \in \mathbb{R}^d \). Let \( z : \mathbb{R} \to \mathbb{R}^d, \alpha \mapsto \alpha x \) be an arc. By definition of homogeneity, \( (F \circ z)(\alpha) = \alpha^k F(x) \) for \( \alpha > 0 \). Taking derivative with respect to \( \alpha \) on both sides (for differentiable points), we have

\[
\forall h \in \partial \alpha F(\alpha x) : \langle x, h \rangle = k \alpha^{k-1} F(x)
\]

(11)

holds for a.e. \( \alpha > 0 \). Pick an arbitrary \( \alpha > 0 \) making (11) hold. Then by (a), (11) is equivalent to \( \forall h \in \partial \alpha F(x) : \langle x, \alpha^{k-1} h \rangle = k \alpha^{k-1} F(x) \), which proves (b).

Applying Theorem A.4 to homogeneous neural networks, we have the following corollary:

**Corollary A.5.** Under the assumptions (A1) and (A2), for any \( \theta \in \mathbb{R}^d \) and \( x \in \mathbb{R}^d \),

\[
(\theta, \partial^\alpha \Phi_x(\theta)) = L \cdot \Phi_x(\theta),
\]

where \( \Phi_x(\theta) = \Phi(\theta; x) \) is the network output for a fixed input \( x \).

Corollary A.5 can be used to derive an exact formula for the weight growth during training.

**Theorem A.6.** For a.e. \( t \geq 0 \),

\[
\frac{1}{2} \frac{d\rho^2}{dt} = L \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n) q_n.
\]

Proof. The proof idea is to use Corollary A.5 and chain rules (See Appendix G for chain rules in Clarke’s sense). Applying the chain rule on \( t \to \rho^2 = \| \theta \|_2^2 \) yields \( \frac{d\rho^2}{dt} = -\langle \theta, h \rangle \) for all \( h \in \partial^\alpha L \) and a.e. \( t > 0 \). Then applying the chain rule on \( \theta \to L \), we have

\[
-\partial^\alpha L \leq \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n) \partial^\alpha q_n = \left\{ \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n) h_n : h_n \in \partial^\alpha q_n \right\}
\]

By Corollary A.5 \( \langle \theta, h_n \rangle = L q_n \), and thus \( \frac{1}{2} \frac{d\rho^2}{dt} = L \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n) q_n \).

For convenience, we define \( \nu(t) := \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n) q_n \) for all \( t \geq 0 \). Then Theorem A.6 can be rephrased as \( \frac{1}{2} \frac{d\rho^2}{dt} = L \nu(t) \) for a.e. \( t \geq 0 \).

**Lemma A.7.** For all \( t > t_0 \),

\[
\nu(t) \geq g \left( \frac{1}{\rho} \frac{g(\log \frac{1}{\rho})}{g'(\log \frac{1}{\rho})} L \right).
\]

Proof. By Lemma A.4 \( q_n \geq g \left( \log \frac{1}{\rho} \right) \) for all \( n \in [N] \). Then by Assumption (B3),

\[
f'(q_n) q_n \geq f'(g \left( \log \frac{1}{\rho} \right)) \cdot g \left( \log \frac{1}{\rho} \right) = \frac{g \left( \log \frac{1}{\rho} \right)}{g'(\log \frac{1}{\rho})}.
\]

19
Combining this with the definitions of \( \nu(t) \) and \( \mathcal{L} \) gives

\[
\nu(t) = \sum_{n=1}^{N} e^{-f(q_{n})} f'(q_{n}) q_{n} \geq \sum_{n=1}^{N} e^{-f(q_{n})} \frac{g(\log \frac{1}{t})}{g'(\log \frac{1}{t})} = \frac{g(\log \frac{1}{t})}{g'(\log \frac{1}{t})} \mathcal{L}.
\]

**Proof for Lemma A.2** Note that \( \frac{d}{dt} \log \rho = \frac{1}{\rho} \frac{d\rho}{dt} = L \frac{\nu(t)}{\rho^2} \) by Theorem A.6. Then it simply follows from Lemma A.7 that \( \frac{d}{dt} \log \rho > 0 \) for a.e. \( t > t_0 \). For the second inequality, we first prove that \( \log \tilde{\gamma} = \log \left( \frac{L^{-1}(\mathcal{L})}{\rho^2} \right) = \log \left( g(\log \frac{1}{t})/\rho^2 \right) \) exists for all \( t \geq t_0 \). \( \mathcal{L}(t) \) is non-increasing with \( t \). So \( \mathcal{L}(t) < e^{-f(b)} \) for all \( t \geq t_0 \). This implies that (1) \( \log \tilde{\gamma} \) is always in the domain of \( g \); (2) \( \rho > 0 \) (otherwise \( \mathcal{L} \geq Ne^{-f(0)} > e^{-f(b)} \), contradicting (B4)). Therefore, \( \tilde{\gamma} := g(\log \frac{1}{t})/\rho^2 \) exists and is always positive for all \( t \geq t_0 \), which proves the existence of \( \log \tilde{\gamma} \).

By the chain rule and Lemma A.7 we have

\[
\frac{d}{dt} \log \tilde{\gamma} = \frac{d}{dt} \left( \log \left( g(\log \frac{1}{t}) \right) - L \log \rho \right) = \frac{g'(\log \frac{1}{t})}{g(\log \frac{1}{t})} \cdot \frac{1}{\mathcal{L}} \cdot \left( -\frac{d\mathcal{L}}{dt} \right) - L^2 \cdot \frac{\nu(t)}{\rho^2} \\
\geq \frac{1}{\nu(t)} \cdot \left( -\frac{d\mathcal{L}}{dt} \right) - L^2 \cdot \frac{\nu(t)}{\rho^2} \\
\geq \frac{1}{\nu(t)} \cdot \left( -\frac{d\mathcal{L}}{dt} \right) - L^2 \cdot \frac{\nu(t)}{\rho^2}.
\]

On the one hand, \( -\frac{d\mathcal{L}}{dt} = \frac{\|d\theta\|^2}{2} \) for a.e. \( t > 0 \) by Lemma G.3 on the other hand, \( L\nu(t) = \langle \theta, \frac{d\theta}{dt} \rangle \) by Theorem A.6. Combining these together yields

\[
\frac{d}{dt} \log \tilde{\gamma} \geq \frac{1}{\nu(t)} \left( \frac{d\theta}{dt} \right)^2 - \left( \frac{d\theta}{dt} \right)^2 \rangle = \frac{1}{\nu(t)} \left\| (I - \hat{\theta}^{\top}) \hat{\theta} \right\|^2.
\]

By the chain rule, \( \frac{d\theta}{dt} = \frac{1}{\rho} (I - \hat{\theta}^{\top}) \frac{d\theta}{dt} \) for a.e. \( t > 0 \). So we have

\[
\frac{d}{dt} \log \tilde{\gamma} \geq \frac{\rho^2}{\nu(t)} \left\| \frac{d\theta}{dt} \right\|^2 = L \left( \frac{d}{dt} \log \rho \right)^{-1} \left\| \frac{d\theta}{dt} \right\|^2.
\]

**A.2 Proof for Lemma A.3**

**Proof for Lemma A.3** By Lemma G.3 and Theorem A.6,

\[
-\frac{d\mathcal{L}}{dt} = \left\| \frac{d\theta}{dt} \right\|^2 \geq \left\langle \theta, \frac{d\theta}{dt} \right\rangle \geq L^2 \cdot \frac{\nu(t)^2}{\rho^2}.
\]

Using Lemma A.7 to lower bound \( \nu \) and replacing \( \rho \) with \( g(\log \frac{1}{t})/\tilde{\gamma} \) by the definition of \( \tilde{\gamma} \), we have

\[
-\frac{d\mathcal{L}}{dt} \geq L^2 \cdot \left( \frac{g(\log \frac{1}{t})}{g'(\log \frac{1}{t})} \right)^2 \cdot \left( \frac{\tilde{\gamma}(t)}{g(\log \frac{1}{t})} \right)^{2/L} \geq L^2 \tilde{\gamma}(t_0)^{2/L} \cdot \frac{g(\log \frac{1}{t})^{2-2/L}}{g'(\log \frac{1}{t})^2} \cdot \mathcal{L},
\]

where the last inequality uses the monotonicity of \( \tilde{\gamma} \). So the following holds for a.e. \( t \geq t_0 \),

\[
\frac{g'(\log \frac{1}{t})^2}{g(\log \frac{1}{t})^{2-2/L}} \cdot \frac{d}{dt} \frac{1}{\mathcal{L}} \geq L^2 \tilde{\gamma}(t_0)^{2/L}.
\]
Integrating on both sides from \(t_0\) to \(t\), we can conclude that
\[
G(1/\mathcal{L}) \geq \frac{1}{2} \gamma^2(t_0)^{2/\mathcal{L}}(t - t_0).
\]
Note that \(1/\mathcal{L}\) is non-decreasing. If \(1/\mathcal{L}\) does not grow to \(+\infty\), then neither does \(G(1/\mathcal{L})\). But the RHS grows to \(+\infty\), which leads to a contradiction. So \(\mathcal{L} \to 0\).

To make \(\mathcal{L} \to 0\), \(q_{\text{min}}\) must converge to \(+\infty\). So \(\rho \to +\infty\). \(\square\)

### B Karush-Kuhn-Tucker Conditions

In this section, we review the definition of Karush-Kuhn-Tucker (KKT) conditions for non-smooth optimization problems following from [Dutta et al., 2013].

Consider the following optimization problem \((P)\) for \(x \in \mathbb{R}^d\):
\[
\min_{x} f(x) \quad \text{s.t.} \quad g_n(x) \leq 0 \quad \forall n \in [N]
\]
where \(f, g_1, \ldots, g_n : \mathbb{R}^d \to \mathbb{R}\) are locally Lipschitz functions. We say that \(x \in \mathbb{R}^d\) is a feasible point of \((P)\) if \(x\) satisfies \(g_n(x) \leq 0\) for all \(n \in [N]\).

**Definition B.1 (KKT Point).** A feasible point \(x\) of \((P)\) is a KKT point if \(x\) satisfies KKT conditions: there exists \(\lambda_1, \ldots, \lambda_N \geq 0\) such that
1. \(0 \in \partial^o f(x) + \sum_{n \in [N]} \lambda_n \partial^o g_n(x)\);
2. \(\forall n \in [N] : \lambda_n g_n(x) = 0\).

It is important to note that a global minimum of \((P)\) may not be a KKT point, but under some regularity assumptions, the KKT conditions become a necessary condition for global optimality. The regularity condition we shall use in this paper is the non-smooth version of Mangasarian-Fromovitz Constraint Qualification (MFCQ) (see, e.g., the constraint qualification (C.Q.5) in [Giorgi et al., 2004]):

**Definition B.2 (MFCQ).** For a feasible point \(x\) of \((P)\), \((P)\) is said to satisfy MFCQ at \(x\) if there exists \(v \in \mathbb{R}^d\) such that for all \(n \in [N]\) with \(g_n(x) = 0\),
\[
\forall h \in \partial^o g_n(x) : \langle h, v \rangle > 0.
\]

Following from [Dutta et al., 2013], we define an approximate version of KKT point, as shown below. Note that this definition is essentially the modified \(\epsilon\)-KKT point defined in their paper, but these two definitions differ in the following two ways: (1) First, in their paper, the subdifferential is allowed to be evaluated in a neighborhood of \(x\), so our definition is slightly stronger; (2) Second, their paper fixes \(\delta = \epsilon^2\), but in our definition we make them independent.

**Definition B.3 (Approximate KKT Point).** For \(\epsilon, \delta > 0\), a feasible point \(x\) of \((P)\) is an \((\epsilon, \delta)\)-KKT point if there exists \(\lambda_n \geq 0, k \in \partial^o f(x), h_n \in \partial^o g_n(x)\) for all \(n \in [N]\) such that
1. \(\left\| k + \sum_{n \in [N]} \lambda_n h_n(x) \right\|_2 \leq \epsilon\);
2. \(\forall n \in [N] : \lambda_n g_n(x) \geq -\delta\).

As shown in [Dutta et al., 2013], \((\epsilon, \delta)\)-KKT point is an approximate version of KKT point in the sense that a series of \((\epsilon, \delta)\)-KKT points can converge to a KKT point. We restate their theorem in our setting:

**Theorem B.4** (Corollary of Theorem 3.6 in [Dutta et al., 2013]). Let \(\{x_k \in \mathbb{R}^d : k \in \mathbb{N}\}\) be a sequence of feasible points of \((P)\), \(\{\epsilon_k > 0 : k \in \mathbb{N}\}\) and \(\{\delta_k > 0 : k \in \mathbb{N}\}\) be two sequences. \(x_k\) is an \((\epsilon_k, \delta_k)\)-KKT point for every \(k\), and \(\epsilon_k \to 0, \delta_k \to 0\). If \(x_k \to x\) as \(k \to +\infty\) and MFCQ holds at \(x\), then \(x\) is a KKT point of \((P)\).
C Convergence to the Max-margin Solution

In this section, we analyze the convergent direction of $\theta$, assuming (A1), (A2), (B3), (B4) as mentioned in Section 5. Recall that for a homogeneous neural network, the optimization problem (P) is defined as follows:

$$\min \quad \frac{1}{2} \|\theta\|_2^2$$
$$\text{s.t.} \quad q_n(\theta) \geq 1 \quad \forall n \in [N]$$

Using the terminologies in Section B, the objective and constraints are $f(\theta) = \frac{1}{2} \|\theta\|_2^2$ and $g_n(\theta) = 1 - q_n(\theta)$ (note that $f$ and $g$ are not the functions defined in (B3)). The KKT conditions for (P) are defined as follows:

**Definition C.1** (KKT Point of (P)). A feasible point $\theta$ of (P) is a KKT point if there exist $\lambda_1, \ldots, \lambda_N \geq 0$ such that

1. $\theta - \sum_{n=1}^N \lambda_n h_n = 0$ for some $h_1, \ldots, h_N$ satisfying $h_n \in \partial^c q_n(\theta);$  
2. $\forall n \in [N] : \lambda_n (q_n(\theta) - 1) = 0.$

**Definition C.2** (Approximate KKT Point of (P)). A feasible point $\theta$ of (P) is an $(\epsilon, \delta)$-KKT point of (P) if there exists $\lambda_1, \ldots, \lambda_N \geq 0$ such that

1. $\|\theta - \sum_{n=1}^N \lambda_n h_n\|_2 \leq \epsilon$ for some $h_1, \ldots, h_N$ satisfying $h_n \in \partial^c q_n(\theta);$  
2. $\forall n \in [N] : \lambda_n (q_n(\theta) - 1) \leq \delta.$

By the homogeneity of $q_n$, it is easy to see that (P) satisfies MFCQ, and thus KKT conditions are first-order necessary condition for global optimality.

**Lemma C.3.** (P) satisfies MFCQ at every feasible point $\theta$.

**Proof.** Take $v := \tilde{\theta}$. For all $n \in [N]$, by homogeneity of $q_n$, we have $\langle v, h \rangle = L q_n(\tilde{\theta}) = L > 0$ holds for any $h \in -\partial^c q_n(\tilde{\theta}) = \partial^c (1 - q_n(\tilde{\theta})).$ \qed

Now we state our results. First, we show the directional convergence of $\tilde{\theta}(t)$ to a KKT point.

**Theorem C.4.** For every limit point $\tilde{\theta}$ of $\{ \tilde{\theta}(t) : t \geq 0 \}$, $\tilde{\theta}/q_{\min}(\tilde{\theta})^{1/L}$ is a KKT point of (P).

Second, we show that after finite time, gradient flow can pass through an approximate KKT point.

**Theorem C.5.** For any $\epsilon, \delta > 0$, there exists $r := \Theta(\log \delta^{-1})$ and $\Delta := \Theta(\epsilon^{-2})$ such that $\theta/q_{\min}(\theta)^{1/L}$ is an $(\epsilon, \delta)$-KKT point at some time $t_*$ satisfying $\log \rho(t_*) \in (r, r + \Delta).$

C.1 Key Ideas

Define $\beta(t) := \frac{1}{\|\partial^c q_n(\theta)\|_2} \langle \tilde{\theta}, \frac{\partial^c q_n(\theta)}{\partial \theta} \rangle$ to be the cosine of the angle between $\theta$ and $\partial^c q_n(\theta)$. Here $\beta(t)$ is only defined for a.e. $t > 0$.

Since $q_n$ is locally Lipschitz, it can be shown that $q_n$ is (globally) Lipschitz on the compact set $S^{d-1}$, which is the unit sphere in $\mathbb{R}^d$. Define

$$B_0 := \sup \left\{ \frac{q_n}{\rho^{L-1}} : \theta \in \mathbb{R}^d \setminus \{0\} \right\}$$
$$= \sup \left\{ q_n : \theta \in S^{d-1} \right\} < \infty.$$

$$B_1 := \sup \left\{ \|h\|_2 : \theta \in \mathbb{R}^d \setminus \{0\}, h \in \partial^c q_n, n \in [N] \right\}$$
$$= \sup \left\{ \|h\|_2 : \theta \in S^{d-1}, h \in \partial^c q_n, n \in [N] \right\} < \infty.$$

22
For showing Theorem C.4 and Theorem C.5 the key idea is to show the following two key lemmas about \( \beta(t) \):

**Lemma C.6.** Let \( C_1, C_2 \) be two constants defined as

\[
C_1 := \frac{\sqrt{2}}{\gamma(t_0)^{1/L}}, \quad C_2 := \frac{2eNK^2}{L\gamma(t_0)^{2/L}} \left( \frac{B_1}{\gamma(t_0)} \right)^{\log_2 K},
\]

where \( K, b_g \) are constants specified in (B3.4). If \( \log \frac{1}{2} \geq b_g \) at time \( t > t_0 \), then \( \tilde{\theta} := \theta/q_{\min}(\theta)^{1/L} \) is an \((\epsilon, \delta)\)-KKT point of \((P)\), where \( \epsilon := C_1\sqrt{1-\beta} \) and \( \delta := C_2/(\log \frac{1}{2}) \).

**Lemma C.7.** For all \( t_2 > t_1 \geq t_0 \),

\[
\int_{t_1}^{t_2} \left( \beta(\tau)^{-2} - 1 \right) \cdot \frac{d}{d\tau} \log \rho(\tau) \cdot d\tau \leq \frac{1}{L} \log \frac{\tilde{\gamma}(t_2)}{\tilde{\gamma}(t_1)}.
\]

In light of Lemma C.6 if we aim to show that \( \theta \) is along the direction of an approximate KKT point, we only need to show \( \beta \to 1 \) (which makes \( \epsilon \to 0 \)) and \( L \to 0 \) (which makes \( \delta \to 0 \)). By Theorem A.1, we have already known that \( L \to 0 \). So it remains to bound \( \beta(t) \).

Since \( \tilde{\gamma}(t) \) is non-decreasing and \( \tilde{\gamma}(t) \leq q_{\min}/\rho^L \leq B_0 \), \( \tilde{\gamma}_\infty := \lim_{t \to +\infty} \tilde{\gamma}(t) \) exists by the monotone convergence theorem. Combining this with Lemma C.7 we can obtain that the integration of \( \beta^{-2} - 1 \) can be upper-bounded. Intuitively, this means that many \( \beta(t) \) are close to 1, and it is easy to make this intuition formal in both asymptotic and non-asymptotic analysis.

### C.2 Proof for Key Lemmas

**Proof of Lemma C.6.** Let \( h(t) := \frac{d\theta}{dt}(t) \) for a.e. \( t > 0 \). By the chain rule, there exist \( h_1, \ldots, h_N \) such that \( h_n \in \partial^o q_n \)

\[
h = \sum_{n \in [N]} e^{-f(q_n)} f'(q_n) h_n.
\]

Let \( h_n := h_n/q_{\min}^{1-1/L} \in \partial^o q_n(\tilde{\theta}) \) (Recall that \( \tilde{\theta} := \theta/q_{\min}(\theta)^{1/L} \)). Construct \( \lambda_n := q_{\min}^{-2/L} \rho \cdot e^{-f(q_n)} f'(q_n)/\|h\|_2 \). Now we only need to show

\[
\left\| \tilde{\theta} - \sum_{n=1}^{N} \lambda_n h_n \right\|_2 \leq \frac{2}{\tilde{\gamma}^{2/L}(1 - \beta)}
\]

(12)

\[
\sum_{n=1}^{N} \lambda_n(q_n(\tilde{\theta}) - 1) \leq \frac{2eNK^2}{L\gamma(t_0)^{2/L}} \left( \frac{B_1}{\gamma(t_0)} \right)^{\log_2 K} \frac{1}{f(\gamma\rho^L)}.
\]

(13)

Then \( \tilde{\theta} \) can be shown to be an \((\epsilon, \delta)\)-KKT point by the monotonicity \( \tilde{\gamma}(t) \geq \tilde{\gamma}(t_0) \) for \( t > t_0 \).

**Proof of (12).** From our construction, \( \sum_{n=1}^{N} \lambda_n h_n = q_{\min}^{-1/L} \rho h/\|h\|_2 \). So

\[
\left\| \tilde{\theta} - \sum_{n=1}^{N} \lambda_n h_n \right\|_2 = q_{\min}^{-2/L} \rho^2 \left\| \tilde{\theta} - \frac{h}{\|h\|_2} \right\|_2 = q_{\min}^{-2/L} \rho^2 (2 - 2\beta) \leq \frac{2}{\tilde{\gamma}^{2/L}(1 - \beta)},
\]

where the last equality is by Lemma C.7.

**Proof for (13).** According to our construction,

\[
\sum_{n=1}^{N} \lambda_n(q_n(\tilde{\theta}) - 1) = q_{\min}^{-2/L} \rho \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n)(q_n - q_{\min}).
\]

Note that \( \|h\|_2 \geq \langle h, \tilde{\theta} \rangle = L\nu/\rho \). By Lemma A.7 and Lemma F.4 we have

\[
\nu \geq \frac{g(\log \frac{1}{2})}{g'(\log \frac{1}{2})} \mathcal{L} \geq \frac{1}{2K} \log \frac{1}{L} \cdot \mathcal{L} \geq \frac{1}{2K} f(\gamma\rho^L)e^{-f(q_{\min})},
\]

23
where the last inequality uses $f(\hat{\gamma}\rho^L) = \log \frac{1}{\hat{\gamma}}$ and $L \geq e^{-f(q_{\text{min}})}$. Combining these gives

$$\lambda_n(q_n(\hat{\theta})) \leq \frac{2Kq_{\text{min}}^{-2/L} \rho^2}{Lf(\hat{\gamma}\rho^L)} \sum_{n=1}^{N} e^{f(q_{\text{min}}) - f(q_n)} f'(q_n)(q_n - q_{\text{min}}).$$

If $q_n > q_{\text{min}}$, then by the mean value theorem there exists $\xi_n \in (q_n, q_{\text{min}})$ such that $f(q_n) - f(q_{\text{min}}) = f'(\xi_n)(q_n - q_{\text{min}})$. By Assumption (B3.4), we know that $f'(q_n) \leq K[\log_2(q_n/\xi_n)] f'(\xi_n)$. Note that $|\log_2(q_n/\xi_n)| \leq \log_2(2B_1\rho^L/q_{\text{min}}) \leq \log_2(2B_1/\hat{\gamma})$. Then we have

$$\lambda_n(q_n(\hat{\theta})) \leq \frac{2Kq_{\text{min}}^{-2/L} \rho^2}{Lf(\hat{\gamma}\rho^L)} \sum_{n: q_n \neq q_{\text{min}}} e^{-f'(\xi_n)(q_n - q_{\text{min}})} f'(\xi_n)(q_n - q_{\text{min}}) \leq \frac{2K\hat{\gamma}^{-2/L} \rho^2}{Lf(\hat{\gamma}\rho^L)} K\log_2(2B_1/\hat{\gamma}).$$

where the second inequality uses $q_{\text{min}}^{-2/L} \rho^2 \leq \hat{\gamma}^{-2/L}$ by Lemma 5.4 and the fact that the function $x \mapsto e^{-x}$ on $(0, +\infty)$ attains the maximum value $e$ at $x = 1$. \hfill \Box

**Proof of Lemma C.7.** By Lemma A.6, $\frac{d}{dt} \log \rho = \frac{1}{\sqrt{\rho}} \frac{d\rho}{dt} = \frac{L\nu}{\sqrt{\rho}}$ for a.e. $t > 0$. By Lemma A.2, for a.e. $t \in (t_1, t_2)$,

$$\frac{d}{dt} \log \hat{\gamma} \geq L \cdot \left\| \frac{\rho^2 \frac{d\theta}{dt}}{L\nu} \right\|_2 \cdot \frac{d}{dt} \log \rho =: \text{RHS}.$$ 

By the chain rule, $\frac{d\theta}{dt} = (\mathbf{I} - \hat{\theta}\hat{\theta}^\top) \frac{d\theta}{dt}$. So we have

$$\text{RHS} = L \cdot \left\| \frac{\rho}{L\nu(t)} (\mathbf{I} - \hat{\theta}\hat{\theta}^\top) \frac{d\theta}{dt} \right\|_2 \cdot \frac{d}{dt} \log \rho$$

$$= L \cdot \left\| \frac{d\theta}{dt} \right\|_2^2 - \left\langle \frac{d\theta}{dt}, \hat{\theta} \right\rangle^2 \cdot \frac{d}{dt} \log \rho$$

$$= L \cdot \beta^{-2} \cdot \frac{d}{dt} \log \rho.$$ 

where the last equality follows from the definition of $\beta$. Integrating on both sides of $\frac{d}{dt} \log \hat{\gamma} \geq L \cdot \beta^{-2} - 1 \cdot \frac{d}{dt} \log \rho$ from $t_1$ to $t_2$ proves the lemma. \hfill \Box

A direct corollary of Lemma C.7 is a very useful upper bound for the minimum $\beta^2 - 1$ within a time interval:

**Corollary C.8.** For all $t_2 > t_1 \geq t_0$, then there exists $t_* \in (t_1, t_2)$ such that

$$\beta(t_*)^{-2} - 1 \leq \frac{1}{L} \cdot \frac{\log \hat{\gamma}(t_2) - \log \hat{\gamma}(t_1)}{\log \rho(t_2) - \log \rho(t_1)).$$

**Proof.** Denote the RHS as $C$. Assume to the contrary that $\beta(\tau)^{-2} - 1 > C$ for a.e. $\tau \in (t_1, t_2)$. By Lemma A.2, $\log \rho(\tau) > 0$ for a.e. $\tau \in (t_1, t_2)$. Then by Lemma C.7 we have

$$\frac{1}{L} \cdot \frac{\log \hat{\gamma}(t_2) - \log \hat{\gamma}(t_1)}{\log \rho(t_2) - \log \rho(t_1))} = \frac{1}{L} \cdot \frac{\log \hat{\gamma}(t_2)}{\gamma(t_1)}.$$ 

which leads to a contradiction. \hfill \Box
C.3 Asymptotic Analysis

To prove Theorem C.4, we consider each limit point \( \theta / q_{\min}(\theta)^{1/L} \) and construct a series of approximate KKT points converging to it. Then \( \theta / q_{\min}(\theta)^{1/L} \) can be shown to be a KKT point by Theorem B.4. The following lemma ensures that such construction exists.

**Lemma C.9.** For every limit point \( \theta \) of \( \{ \theta(t) : t \geq 0 \} \), there exists a sequence of \( \{ t_m : m \in \mathbb{N} \} \) such that \( t_m \to +\infty, \theta(t_m) \to \bar{\theta} \), and \( \beta(t_m) \to 1 \).

Before proving Lemma C.9, we first prove an auxiliary lemma.

**Lemma C.10.** For a.e. \( t > t_0 \),

\[
\left\| \frac{d\theta}{dt} \right\|_2 \leq \frac{d}{dt} \log \rho.
\]

**Proof.** Observe that
\[
\left\| \frac{d\theta}{dt} \right\|_2 = \frac{1}{\rho} \left\| (I - \partial \bar{\theta}^T) \frac{d\theta}{dt} \right\|_2 \leq \frac{1}{\rho} \left\| \frac{d\theta}{dt} \right\|_2.
\]

By the chain rule, there exists \( h_1, \ldots, h_N : \mathbb{R}_{\geq 0} \to \mathbb{R}^d \) satisfying that for a.e. \( t > 0 \), \( h_n(t) \in \partial^p q_n \) and \( \frac{d\theta}{dt} = \sum_{n \in [N]} e^{-f(q_n)} f'(q_n) h_n(t) \).

By definition of \( B_1, \| h_n \|_2 \leq B_1 \rho^{L-1} \) for a.e. \( t > 0 \). So we have
\[
\left\| \frac{d\theta}{dt} \right\|_2 \leq \sum_{n \in [N]} e^{-f(q_n)} f'(q_n) \| h_n \|_2 \\
\leq \sum_{n \in [N]} e^{-f(q_n)} f'(q_n) q_n \cdot \frac{1}{q_n} \cdot B_1 \rho^{L-1}.
\]

Note that every summand is positive. By Lemma 5.4, \( q_n \) is lower-bounded by \( q_n \geq q_{\min} \geq g(\log \frac{1}{\rho}) \), so we can replace \( q_n \) with \( g(\log \frac{1}{\rho}) \) in the above inequality. Combining with the fact that \( \sum_{n \in [N]} e^{-f(q_n)} f'(q_n) q_n \) is just \( \nu \), we have
\[
\left\| \frac{d\theta}{dt} \right\|_2 \leq \frac{\nu}{g(\log \frac{1}{\rho})} \cdot B_1 \rho^{L-1}.
\]

By the definition of \( \bar{\gamma} \) and Lemma A.6, the RHS can be rewritten as
\[
\frac{\nu}{g(\log \frac{1}{\rho})} \cdot B_1 \rho^{L-1} = \frac{B_1 \nu}{\bar{\gamma}} = \frac{B_1}{\bar{\gamma} L} \cdot d \log \rho.
\]

So we have
\[
\left\| \frac{d\theta}{dt} \right\|_2 \leq \frac{1}{\rho} \left\| \frac{d\theta}{dt} \right\|_2 \leq \frac{B_1}{\bar{\gamma} L} \cdot \frac{d}{dt} \log \rho.
\]

**Proof of Lemma C.9.** Let \( \{ \epsilon_m : m \in \mathbb{N} \} \) be an arbitrary sequence with \( \epsilon_m \to 0 \). Now we construct \( \{ t_m \} \) by induction. Suppose \( t_1 < t_2 < \cdots < t_{m-1} \) have already been constructed. Since \( \theta \) is a limit point and \( \bar{\gamma}(t) \uparrow \bar{\gamma}_\infty \) (recall that \( \bar{\gamma}_\infty := \lim_{t \to +\infty} \bar{\gamma}(t) \)), there exists \( s_m > t_{m-1} \) such that
\[
\left\| \bar{\theta}(s_m) - \bar{\theta} \right\|_2 \leq \epsilon_m \quad \text{and} \quad \frac{1}{L} \log \frac{\bar{\gamma}_\infty}{\bar{\gamma}(s_m)} \leq \epsilon_m^3.
\]

Let \( s'_m > s_m \) be a time such that \( \log \rho(s'_m) = \log \rho(s_m) + \epsilon_m \). According to Theorem A.1, \( \log \rho \to +\infty \), so \( s'_m \) must exist. We construct \( t_m \in (s_m, s'_m) \) to be a time that \( \beta(t_m)^{-2} - 1 \leq \epsilon_m^2 \), where the existence can be shown by Corollary C.3.

Now we show that this construction meets our requirement. It follows from \( \beta(t_m)^{-2} - 1 \leq \epsilon_m^2 \) that \( \beta(t_m) \geq 1/\sqrt{1 + \epsilon_m^2} \to 1 \). By Lemma C.10, we also know that
\[
\left\| \theta(t_m) - \bar{\theta} \right\|_2 \leq \frac{B_1}{L \bar{\gamma}(t_0)} \cdot \epsilon_m + \epsilon_m \to 0.
\]

This completes the proof. \( \square \)
Proof of Theorem C.4 Let \( \hat{\theta} := \theta / q_{\min}(\theta)^{1/L} \) for short. Let \( \{ t_m : m \in \mathbb{N} \} \) be the sequence constructed as in Lemma C.9. For each \( t_m \), define \( \epsilon(t_m) \) and \( \delta(t_m) \) as in Lemma C.6. Then we know that \( \hat{\theta}(t_m) / q_{\min}(t_m)^{1/L} \) is an \((\epsilon(t_m), \delta(t_m))\)-KKT point and \( \hat{\theta}(t_m) / q_{\min}(t_m)^{1/L} \rightarrow \theta, \epsilon(t_m) \rightarrow 0, \delta(t_m) \rightarrow 0 \). By Lemma C.3 (P) satisfies MFCQ. Then applying Theorem B.4 proves this theorem.

C.4 Non-asymptotic Analysis

Proof of Theorem C.5 Let \( C_0 := \frac{1}{L} \log \frac{\epsilon_m}{\gamma(t_0)} \). Without loss of generality, we assume \( \epsilon < \sqrt{2\frac{E}{S}} C_1 \) and \( \delta < C_2 / f(b_f) \). Let \( t_1 \) be the time such that \( \log \rho(t_1) = \frac{1}{L} \log \frac{2(C_2 \delta^{-1})}{\gamma(t_0) \delta} = \Theta(\log \delta^{-1}) \) and \( t_2 \) be the time such that \( \log \rho(t_2) - \log \rho(t_1) = \frac{1}{2} C_0 C_2^2 \delta^{-2} = \Theta(\epsilon^{-2}) \). By Corollary C.8 there exists \( t_s \in (t_1, t_2) \) such that \( \beta(t_s)^{-2} - 1 \leq 2\epsilon C_1^{-2} \).

Now we argue that \( \hat{\theta}(t_s) \) is an \((\epsilon, \delta)\)-KKT point. By Lemma C.6, we only need to show \( C_1 \sqrt{1 - \beta(t_s)} \leq \epsilon \) and \( C_2 / f(\gamma(t_s) \rho(t_s)^L) \leq \delta \). For the first inequality, by assumption \( \epsilon < \sqrt{2\frac{E}{S}} C_1 \), we know that \( \beta(t_s)^{-2} \leq -1 < 3 \), which implies \( |\beta(t_s)| < \frac{1}{2} \). Then we have \( 1 - \beta(t_s) \leq 2\epsilon C_1^{-2} \), \( \sqrt{\frac{\epsilon(t_s)^2}{\rho^L}} \leq C_2 \epsilon C_1^{-2} \). Therefore, \( C_1 \sqrt{1 - \beta(t_s)} \leq \epsilon \) holds. For the second inequality, \( \gamma(t_s) \rho(t_s)^L \geq \tilde{\gamma}(t_s) \cdot \frac{g(C_2 \delta^{-1})}{\gamma(t_0)} \geq g(C_2 \delta^{-1}) \). Therefore, \( C_2 / f(\tilde{\gamma}(t_s) \rho(t_s)^L) \leq \delta \) holds.

D Extension: Multi-class Classification

In this section, we generalize our results to multi-class classification with cross-entropy loss. This part of analysis is inspired by Theorem 1 in [Zhang et al., 2019], which gives a lower bound for the gradient in terms of the loss \( \mathcal{L} \).

Since now a neural network has multiple outputs, we need to redefine our notations. Let \( C \) be the number of classes. The output of a neural network \( \Phi \) is a vector \( \Phi(\theta; x) \in \mathbb{R}^C \). We use \( \Phi_j(\theta; x) \in \mathbb{R} \) to denote the \( j \)-th output of \( \Phi \) on the input \( x \in \mathbb{R}^{d_x} \). A dataset is denoted by \( D = \{x_n, y_n \}_{n=1}^N = \{(x_n, y_n) : n \in [N]\} \), where \( x_n \in \mathbb{R}^{d_x} \) is a data input and \( y_n \in [C] \) is the corresponding label. The loss function of \( \Phi \) on the dataset \( D \) is defined as

\[
\mathcal{L}(\theta) := \sum_{n=1}^N -\log \frac{e^{-\Phi_{y_n}(\theta; x_n)}}{\sum_{j=1}^C e^{-\Phi_j(\theta; x_n)}}.
\]

The margin for a single data point \((x_n, y_n)\) is defined to be \( q_n(\theta) := \Phi_{y_n}(\theta; x_n) - \max_{j \neq y_n} \{\Phi_j(\theta; x_n)\} \), and the margin for the entire dataset is defined to be \( q_{\min}(\theta) = \min_{n \in [N]} q_n(\theta) \). We define the normalized margin to be \( \tilde{\gamma}(\theta) := q_{\min}(\theta) / \rho^L \), where \( \rho := \|\theta\|_2 \) and \( \theta := \theta / \rho \in \mathcal{S}^{d-1} \) as usual.

Let \( \ell(q) := \log(1 + e^{-q}) \) be the logistic loss. Recall that \( \ell(q) \) satisfies (B3). Let \( f(q) = -\log \ell(q) = -\log(1 + e^{-q}) \). Let \( g \) be the inverse function of \( f \). So \( g(q) = -\log(e^{e^{-q}} - 1) \).

The cross-entropy loss can be rewritten in other ways. Let \( s_{nj} := \Phi_{y_n}(\theta; x_n) - \Phi_j(\theta; x_n) \). Let \( \tilde{q}_n := -\text{LSE}(-s_{nj} : j \neq y_n) = -\log \left( \sum_{j \neq y_n} e^{-s_{nj}} \right) \). Then

\[
\mathcal{L}(\theta) := \sum_{n=1}^N \log \left( 1 + \sum_{j \neq y_n} e^{-s_{nj}} \right) = \sum_{n=1}^N \log(1 + e^{-\tilde{q}_n}) = \sum_{n=1}^N \ell(\tilde{q}_n) = \sum_{n=1}^N e^{-f(\tilde{q}_n)}.
\]

(M1). (Regularity). For any fixed \( x \), for every \( j \in [C] \), \( \Phi_j(\cdot; x) \) is locally Lipschitz and admits a chain rule;

(M2). (Homogeneity). For any fixed \( x \), for every \( j \in [C] \), \( \Phi_j(\cdot; x) \) is \( L \)-homogeneous;

(M3). (Cross-entropy Loss). \( \mathcal{L}(\theta) \) is defined as the cross-entropy loss on the training set.

(M4). (Separability). There exists a time \( t_0 \) such that \( \mathcal{L}(t_0) < \log 2 \).
If $L < \log 2$, then $\sum_{j \neq y_n} e^{-s_{nj}} < 1$ for all $n \in [N]$, and thus $s_{nj} > 0$ for all $n \in [N], j \in [C]$. So (M4) ensures the separability of training data.

**Definition D.1.** For cross-entropy loss, the smoothed normalized margin $\tilde{\gamma}(\theta)$ of $\theta$ is defined as

$$\tilde{\gamma}(\theta) := \frac{\ell^{-1}(L)}{\rho^L} = -\log(e^L - 1),$$

where $\ell^{-1}(\cdot)$ is the inverse function of the logistic loss $\ell(\cdot)$.

Theorem 4.2 and 4.3 still hold. Here we redefine the optimization problem (P) to be

$$\min \frac{1}{2} \|\theta\|_2^2 \quad \text{s.t.} \quad s_{nj}(\theta) \geq 1 \quad \forall n \in [N], \forall j \in [C] \setminus \{y_n\}$$

Most of our proofs are very similar as before. Here we only show the proof for the generalized version of Lemma 4.4.

**Lemma D.2.** Lemma 4.4 is also true for the smoothed normalized margin $\tilde{\gamma}$ defined in Definition D.1.

**Proof.** Define $\nu(t)$ by the following formula:

$$\nu(t) := \sum_{n=1}^N \sum_{j \neq y_n} e^{-s_{nj}} s_{nj}.$$

Using a similar argument as in Theorem A.6, it can be proved that $\frac{d\nu}{dt} = L \nu(t)$ for $t > 0$.

It can be shown that Lemma A.7 which asserts that $\nu(t) \geq \frac{g(\log \frac{1}{2})}{g'(\log \frac{1}{2})} L$, still holds for this new definition of $\nu(t)$. By definition, $s_{nj} \geq q_n$. Also note that $e^{-\tilde{q}_n} \geq e^{-q_n}$. So $s_{nj} \geq q_n \geq \tilde{q}_n$. Then we have

$$\nu(t) \geq \sum_{n=1}^N \sum_{j \neq y_n} e^{-s_{nj}} \cdot \tilde{q}_n = \sum_{n=1}^N e^{-\tilde{q}_n} \cdot \tilde{q}_n = \sum_{n=1}^N e^{-f(\tilde{q}_n) f'(\tilde{q}_n) \tilde{q}_n}.$$

Note that $L = \sum_{n=1}^N e^{-f(\tilde{q}_n)}$. Then using Lemma A.7 for logistic loss can conclude that $\nu(t) \geq \frac{g(\log \frac{1}{2})}{g'(\log \frac{1}{2})} L$.

The rest of the proof for this lemma is exactly as same as that for Lemma 4.4.

---

**E Extension: Multi-homogeneous Models**

In this section, we extend our results to multi-homogeneous models. Let $\Phi(w_1, \ldots, w_m; x)$ be $(k_1, \ldots, k_m)$-homogeneous. Let $\rho_i = \|w_i\|_2$ and $\tilde{w}_i = \frac{w_i}{\|w_i\|_2}$.

The smoothed normalized margin defined in (8) can be rewritten as follows:

**Definition E.1.** For a multi-homogeneous model with loss function $\ell(\cdot)$ satisfying (B3), the smoothed normalized margin $\tilde{\gamma}(\theta)$ of $\theta$ is defined as

$$\tilde{\gamma}(\theta) = \frac{g(\log \frac{1}{2})}{\prod_{i=1}^m \rho_i} = \frac{\ell^{-1}(L)}{\prod_{i=1}^m \rho_i^L}.$$ 

We only prove the generalized version of Lemma 4.4 here. The other proofs are almost the same.

**Lemma E.2.** For all $t > t_0$,

$$\frac{d}{dt} \log \rho_i > 0 \quad \text{for all } i \in [m] \quad \text{and} \quad \frac{d}{dt} \log \tilde{\gamma} \geq \sum_{i=1}^m k_i \left( \frac{d}{dt} \log \rho_i \right)^{-1} \left\| \frac{d\tilde{w}_i}{dt} \right\|_2^2.$$ (14)
Proof. Note that \( \frac{d}{dt} \log \rho_i = \frac{1}{2\rho_i^2} \frac{d\nu(t)}{dt} \) by Theorem A.6.
It simply follows from Lemma A.7 that \( \frac{d}{dt} \log \rho > 0 \) for a.e. \( t > t_0 \).
And it is easy to see that \( \log \gamma = \log \left( \frac{g(\log \frac{1}{\rho})}{\rho^g} \right) \) exists for all \( t \geq t_0 \).
By the chain rule and Lemma A.7, we have
\[
\frac{d}{dt} \log \gamma = \frac{d}{dt} \left( \log \left( \frac{g(\log \frac{1}{\rho})}{g(\log \frac{1}{\rho})} \right) - \sum_{i=1}^{m} k_i \log \rho_i \right) = \frac{g'(\log \frac{1}{\rho})}{g(\log \frac{1}{\rho})} \cdot \frac{1}{\rho} \left( - \frac{d\mathcal{L}}{dt} - \sum_{i=1}^{m} \frac{k_i^2 \nu(t)}{\rho_i^2} \right)
\geq \frac{1}{\nu(t)} \cdot \left( - \frac{d\mathcal{L}}{dt} - \sum_{i=1}^{m} \frac{k_i^2 \nu(t)}{\rho_i^2} \right)
\geq \frac{1}{\nu(t)} \cdot \left( - \frac{d\mathcal{L}}{dt} - \frac{m}{\rho^2} \right).
\]

On the one hand, \( \frac{d\mathcal{L}}{dt} = \sum_{i=1}^{m} \left\| \frac{dw_i}{dt} \right\|^2_2 \) for a.e. \( t > 0 \) by Lemma G.3.
On the other hand, \( k_i \nu(t) = \langle w_i, \frac{dw_i}{dt} \rangle \) by Theorem A.6.
Combining these together yields
\[
\frac{d}{dt} \log \gamma \geq \frac{1}{\nu(t)} \sum_{i=1}^{m} \left( \left\| \frac{dw_i}{dt} \right\|^2_2 - \langle w_i, \frac{dw_i}{dt} \rangle^2 \right) = \frac{1}{\nu(t)} \left\| (I - \hat{w}_i \hat{w}_i^\top) \frac{dw_i}{dt} \right\|^2_2.
\]

By the chain rule, \( \frac{d\hat{w}_i}{dt} = \frac{1}{\rho_i} (I - \hat{w}_i \hat{w}_i^\top) \frac{dw_i}{dt} \) for a.e. \( t > 0 \).
So we have
\[
\frac{d}{dt} \log \gamma \geq \sum_{i=1}^{m} \rho_i^2 \left\| \frac{d\hat{w}_i}{dt} \right\|^2_2 = \sum_{i=1}^{m} k_i \left( \frac{d}{dt} \log \rho_i \right)^{-1} \left\| \frac{d\hat{w}_i}{dt} \right\|^2_2.
\]

For cross-entropy loss, we can combine the proofs in Appendix D to show that Lemma E.2 holds if we use the following definition of the smoothed normalized margin:

**Definition E.3.** For a multi-homogeneous model with cross-entropy, the smoothed normalized margin \( \tilde{\gamma}(\theta) \) of \( \theta \) is defined as
\[
\tilde{\gamma} = \frac{\ell^{-1}(\mathcal{L})}{\prod_{i=1}^{m} \rho_i^{k_i}} = \frac{-\log(e^\mathcal{L} - 1)}{\prod_{i=1}^{m} \rho_i^{k_i}}.
\]
where \( \ell^{-1}(\cdot) \) is the inverse function of the logistic loss \( \ell(\cdot) \).

The only place we need to change in the proof for Lemma E.2 is that instead of using Lemma A.7, we need to prove \( \nu(t) \geq \frac{g(\log \frac{1}{\rho})}{g(\log \frac{1}{\rho})} \mathcal{L} \) in a similar way as in Lemma D.2.
The other parts of the proof are exactly the same as before.

## F Tight Bounds for Loss Convergence and Weight Growth

In this section, we give tight bounds for loss convergence and weight growth under Assumption (A1), (A2), (B3), (B4).
The main theorem is stated below:

**Theorem F.1.** For \( t > t_0 \), we have
\[
\mathcal{L}(t) = \Theta \left( \frac{g(\log t)^{2/L}}{t(\log t)^2} \right) \quad \text{and} \quad \rho = \Theta((\log t)^{1/L}).
\]

Applying Theorem F.1 to exponential loss and logistic loss, in which \( g(x) = \Theta(x) \), we have the following corollary:

**Corollary F.2.** If \( \ell(\cdot) \) is the exponential or logistic loss, then for \( t > t_0 \),
\[
\mathcal{L}(t) = \Theta \left( \frac{1}{t(\log t)^2} \right) \quad \text{and} \quad \rho = \Theta((\log t)^{1/L}).
\]
The key idea to prove Theorem F.1 is to utilize Lemma A.3 in which \( \mathcal{L}(t) \) is bounded from above by \( \frac{1}{G^{-1}(t)} \). So upper bounding \( \mathcal{L}(t) \) reduces to lower bounding \( G^{-1} \). In the following lemma, we obtain tight asymptotic bounds for \( G(\cdot) \) and \( G^{-1}(\cdot) \):

**Lemma F.3.** For function \( G(\cdot) \) defined in Lemma A.3 and its inverse function \( G^{-1}(\cdot) \), we have the following bounds:

\[
G(x) = \Theta \left( \frac{g(\log x)^{2/L}}{(\log x)^2 - x} \right) \quad \text{and} \quad G^{-1}(y) = \Theta \left( \frac{(\log y)^2}{g(\log y)^{2/L}} \right).
\]

For other bounds, we derive them as follows. We first show that \( g(\log \frac{1}{t}) = \Theta(\rho^L) \). With this equivalence, we derive an upper bound for the gradient at each time \( t \) in terms of \( \mathcal{L} \), and take an integration to bound \( \mathcal{L}(t) \) from below. Now we have both lower and upper bounds for \( \mathcal{L}(t) \). Plugging these two bounds to \( g(\log \frac{1}{t}) = \Theta(\rho^L) \) gives the lower and upper bounds for \( \rho(t) \).

**F.1 Consequences of (B3.4)**

Before proving Theorem F.1, we show some consequences of (B3.4).

**Lemma F.4.** For \( f(\cdot) \) and \( g(\cdot) \), we have

1. For all \( x \in [b_y, +\infty) \), \( \frac{g(x)}{g'(x)} \in \left[ \frac{1}{2Kx}, 2Kx \right] \);
2. For all \( y \in [g(b_y), +\infty) \), \( \frac{f(y)}{f'(y)} \in \left[ \frac{1}{2Ky}, 2Ky \right] \).

Thus, \( g(x) = \Theta(xg'(x)) \), \( f(y) = \Theta(yf'(y)) \).

**Proof.** To prove Item 1, it is sufficient to show that

\[
g(x) = b_f + \int_{f(b_f)}^{x} g'(u)du \geq \int_{x/2}^{x} g'(u)du \]

\[
\geq (x/2) \cdot \frac{g'(x)}{K} = \frac{1}{2K} \cdot x g'(x).
\]

\[
x = f(b_f) + \int_{f(b_f)}^{x} g'(u)f'(g(u))du \geq \int_{f(g(b_y)/2)}^{x} g'(u)du
\]

\[
= (x/2) \cdot \frac{f'(g(x))}{K} = \frac{1}{2K} \cdot f'(g(x)).
\]

To prove Item 2, we only need to notice that Item 1 implies \( yf'(y) = \frac{g(f(y))}{g'(f(y))} \in \left[ \frac{1}{2Kf(y)}, 2Kf(y) \right] \) for all \( y \in [g(b_y), +\infty) \).

Recall that (B3.4) directly implies that \( f'(\Theta(x)) = \Theta(f'(x)) \) and \( g'(\Theta(x)) = \Theta(g'(x)) \). Combining this with Lemma F.4, we have the following corollary:

**Corollary F.5.** \( f(\Theta(x)) = \Theta(f(x)) \) and \( g(\Theta(x)) = \Theta(g(x)) \).

Also, note that Lemma F.4 essentially shows that \( (\log f(x))' = \Theta(1/x) \) and \( (\log g(x))' = \Theta(1/x) \). So \( \log f(x) = \Theta(\log x) \) and \( \log g(x) = \Theta(\log x) \), which means that \( f \) and \( g \) grow at most polynomially.

**Corollary F.6.** \( f(x) = x^{\Theta(1)} \) and \( g(x) = x^{\Theta(1)} \).
F.2 Proof for Theorem F.1

We use the notations $B_0, B_1$ from Appendix C.1. In this section, we first prove Lemma F.3 to obtain tight bounds for $G(\cdot)$ and $G^{-1}(\cdot)$, and then use them to prove Theorem F.1.

Proof for Lemma F.3 We first prove the bounds for $G(x)$, and then prove the bounds for $G^{-1}(y)$.

Bounding for $G(x)$. Let $C_G = \int_{\exp(b_y)}^{\infty} g'(\log u)^2 \frac{1}{g(\log u)^2} du$. For $x \geq \exp(b_y)$,

$$G(x) = C_G + \int_{\exp(b_y)}^{x} g'(\log u)^2 \frac{1}{g(\log u)^2} du$$

$$= C_G + \int_{\exp(b_y)}^{x} \left( \frac{g'(\log u) \log u}{g(\log u)} \right)^2 \frac{g(\log u)^2}{(\log u)^2} du.$$ 

By Lemma F.4

$$G(x) \leq C_G + 4K^2 g(\log x)^{2/L} \int_{\exp(b_y)}^{x} \frac{1}{(\log u)^2} du = O \left( \frac{g(\log x)^{2/L}}{(\log x)^2} x \right).$$

On the other hand, for $x \geq \exp(2b_y)$, we have

$$G(x) \geq \frac{1}{4K^2} \int_{\sqrt{\pi}}^{x} \frac{g(\log u)^{2/L}}{(\log u)^2} du \geq \frac{1}{4K^2} g((\log x)/2)^{2/L} \int_{\sqrt{\pi}}^{x} \frac{1}{(\log u)^2} du = \Omega \left( \frac{g(\log x)^{2/L}}{(\log x)^2} x \right).$$

Bounding for $G^{-1}(y)$. Let $x = G^{-1}(y)$ for $y \geq 0$. $G(x)$ always has a finite value whenever $x$ is finite. So $x \rightarrow +\infty$ when $y \rightarrow +\infty$. According to the first part of the proof, we know that $y = \Theta \left( \frac{g(\log x)^{2/L}}{(\log x)^2} x \right)$. Taking logarithm on both sides and using Corollary F.6, we have $\log y = \Theta(\log x)$. By Corollary F.5 $g(\log y) = g(\Theta(\log x)) = \Theta(g(\log x))$. Therefore,

$$\frac{(\log y)^2}{g(\log y)^{2/L} y} = \Theta \left( \frac{(\log x)^2}{g(\log x)^{2/L} y} \right) = \Theta(x).$$

This implies that $x = \Theta \left( \frac{(\log y)^2}{g(\log y)^{2/L} y} \right).$ \qquad $\Box$

Proof for Theorem F.1 We first prove the upper bound for $L$. Then we derive lower and upper bounds for $\rho$ in terms of $L$, and use these bounds to give a lower bound for $L$. Finally, we plug in the tight bounds for $L$ to obtain the lower and upper bounds for $\rho$ in terms of $t$.

Upper Bounding $L$. By Lemma A.3 we have $\frac{1}{L} \geq G^{-1}(\Omega(t))$. Using Lemma F.3 we have $\frac{1}{L} = \Omega \left( \frac{(\log t)^2}{g(\log t)^{2/L} t} \right)$, which completes the proof.

Bounding $\rho$ in Terms of $L$. $\bar{\gamma}(t) \geq \gamma(t_0)$, so $\rho^L \leq \frac{1}{\bar{\gamma}(t_0)} g(\log \frac{1}{L}) = O(g(\log \frac{1}{L})).$ On the other hand, $g(\log \frac{1}{L}) \leq q_{min} \leq B_0 \rho^L$. So $\rho^L = \Omega(g(\log \frac{1}{L})).$ Therefore, we have the following relationship between $\rho^L$ and $g(\log \frac{1}{L})$:

$$\rho^L = \Theta(g(\log \frac{1}{L})).$$ \hfill (15)

Lower Bounding $L$. Let $h_1, \ldots, h_N$ be a set of vectors such that $h_n \in \partial q_n / \partial \theta$ and

$$\frac{d\theta}{dt} = \sum_{n=1}^{N} e^{-f(q_n)} f'(q_n) h_n.$$
By \eqref{eq:derivative} and Corollary \ref{corollary:derivative}, \( f'(q_n) \leq f'(B_0 \rho^L) = f'(O(g(\log \frac{1}{L}))) = O(f'(\log \frac{1}{L}))) = O(1/g'(\log \frac{1}{L})). \) Again by \eqref{eq:derivative}, we have \( \|h_n\|_2 \leq B_1 \rho^{L-1} = O(g(\log \frac{1}{L})^{-\alpha}). \) Combining these two bounds together, it follows from Corollary \ref{corollary:derivative} that
\[
  f'(q_n) \|h_n\|_2 = O \left( \frac{g(\log \frac{1}{L})^{1-\alpha}}{g'(\log \frac{1}{L})} \right) = O \left( \frac{\log \frac{1}{L}}{g'(\log \frac{1}{L})^{1/2L}} \right).
\]

Thus,
\[
  - \frac{dL}{dt} = \left\| \frac{d\theta}{dt} \right\|_2^2 = \left\| \sum_{n=1}^N e^{-f(q_n)} f'(q_n) h_n \right\|_2^2 \leq \left( \sum_{n=1}^N e^{-f(q_n)} \cdot \max_{n \in [N]} \{ f'(q_n) \|h_n\|_2 \} \right)^2 \leq L^2 \cdot O \left( \frac{(\log \frac{1}{L})^2}{g'(\log \frac{1}{L})^{2/2L}} \right).
\]

By definition of \( G(\cdot) \), this implies that there exists a constant \( c \) such that \( \frac{dL}{dt} \leq c \) for any \( L \) that is small enough. We can complete our proof by applying Lemma \ref{lemma:derivative}.

**Bounding \( \rho \) in Terms of \( t \).** By \eqref{eq:derivative} and the tight bound for \( L(t), \rho^L = \Theta(\log \frac{1}{L})) = \Theta(\Theta(\log t)). \) Using Corollary \ref{corollary:derivative} we can conclude that \( \rho^L = \Theta(\log t) \).

\[ \square \]

### G Chain Rules for Non-differentiable Functions

In this section, we provide some background on the chain rule for non-differentiable functions. The ordinary chain rule for differentiable functions is a very useful formula for computing derivatives in calculus. However, for non-differentiable functions, it is difficult to find a natural definition of subdifferential so that the chain rule equation holds exactly. To solve this issue, Clarke proposed Clarke’s subdifferential \cite{Clarke1975,Clarke1990,Clarke2008} for locally Lipschitz functions, for which the chain rule holds as an inclusion rather than an equation:

**Theorem G.1** (Theorem 2.3.9 and 2.3.10 of \cite{Clarke1990}). Let \( z_1, \ldots, z_n : \mathbb{R}^d \to \mathbb{R} \) and \( f : \mathbb{R}^n \to \mathbb{R} \) be locally Lipschitz functions. Let \( (f \circ z)(x) = f(z_1(x), \ldots, z_n(x)) \) be the composition of \( f \) and \( z \). Then,
\[
  \partial^\circ (f \circ z)(x) \subseteq \text{conv} \left\{ \sum_{i=1}^n \alpha_i h_i : \alpha \in \partial^\circ f(z_1(x), \ldots, z_n(x)), h_i \in \partial^\circ z_i(x) \right\}.
\]

For analyzing gradient flow, the chain rule is crucial. For a differentiable loss function \( \mathcal{L}(\theta) \), we can see from the chain rule that the function value keeps decreasing along the gradient flow \( \frac{d\theta(t)}{dt} = -\nabla \mathcal{L}(\theta(t)) \):

\[
  \frac{d\mathcal{L}(\theta(t))}{dt} = \left\langle \nabla \mathcal{L}(\theta(t)), \frac{d\theta(t)}{dt} \right\rangle = -\left\| \frac{d\theta(t)}{dt} \right\|_2^2.
\]

But for locally Lipschitz functions which could be non-differentiable, \eqref{eq:chain} may not hold in general since Theorem G.1 only holds for an inclusion.

Following \cite{Davis2019,Drusvyatskiy2015}, we consider the functions that admit a chain rule for any arc.

**Definition G.2** (Chain Rule). A locally Lipschitz function \( f : \mathbb{R}^d \to \mathbb{R} \) admits a chain rule if for any arc \( z : \mathbb{R}_{\geq 0} \to \mathbb{R}^d \),
\[
  \forall h \in \partial^\circ f(z(t)) : (f \circ z)'(t) = \langle h, z'(t) \rangle
\]
holds for a.e. \( t > 0 \).

It is shown in \cite{Davis2019,Drusvyatskiy2015} that a generalized version of \eqref{eq:chain} holds for such functions:
Lemma G.3 (Lemma 5.2 [Davis et al., 2019]). Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be a locally Lipschitz function that admits a chain rule. Let $\theta : \mathbb{R}_{\geq 0} \to \mathbb{R}^d$ be the gradient flow on $\mathcal{L}$:

$$\frac{d\theta(t)}{dt} \in -\partial \mathcal{L}(\theta(t)) \quad \text{for a.e. } t > 0,$$

Then

$$\frac{d\mathcal{L}(\theta(t))}{dt} = -\left\| \frac{d\theta(t)}{dt} \right\|^2 = -\text{dist}(0, \partial \mathcal{L}(\theta(t)))^2$$

holds for a.e. $t > 0$.

We can see that $C^1$-smooth functions admit chain rules. As shown in [Davis et al., 2019], if a locally Lipschitz function is subdifferentially regular or Whitney $C^1$-stratifiable, then it admits a chain rule. The latter one includes a large family of functions, e.g., semi-algebraic functions, semi-analytic functions, and definable functions in an o-minimal structure [Coste, 2002, van den Dries and Miller, 1996].

It is worth noting that the class of functions that admits chain rules is closed under composition. This is indeed a simple corollary of Theorem G.1.

Theorem G.4. Let $z_1, \ldots, z_n : \mathbb{R}^d \to \mathbb{R}$ and $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz functions and assume all of them admit chain rules. Let $(f \circ z)(x) = f(z_1(x), \ldots, z_n(x))$ be the composition of $f$ and $z$. Then $f \circ z$ also admits a chain rule.

Proof. We can see that $f \circ z$ is locally Lipschitz. Let $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^d, t \mapsto x(t)$ be an arc. First, we show that $z \circ x : \mathbb{R}_{\geq 0} \to \mathbb{R}^d, t \mapsto z(x(t))$ is also an arc. For any closed sub-interval $I$, $z(x(I))$ must be contained in a compact set $U$. Then it can be shown that the locally Lipschitz continuous function $z$ is (globally) Lipschitz continuous on $U$. By the fact that the composition of a Lipschitz continuous and an absolutely continuous function is absolutely continuous, $z \circ x$ is absolutely continuous on $I$, and thus it is an arc.

Since $f$ and $z$ admit chain rules on arcs $z \circ x$ and $x$ respectively, the following holds for a.e. $t > 0$,

$$\forall \alpha \in \partial f(z(x(t))) : \quad (f \circ (z \circ x))'(t) = \langle \alpha, (z \circ x)'(t) \rangle,$$

$$\forall h_i \in \partial z_i(x(t)) : \quad (z_i \circ x)'(t) = \langle h_i, x'(t) \rangle.$$

Combining these we obtain that for a.e. $t > 0$,

$$(f \circ z \circ x)'(t) = \sum_{i=1}^n \alpha_i \langle h_i, x'(t) \rangle,$$

for all $\alpha \in \partial f(z(x(t)))$ and for all $h_i \in \partial z_i(x(t))$. The RHS can be rewritten as $\langle \sum_{i=1}^n \alpha_i h_i, x'(t) \rangle$. By Theorem G.1 every $k \in \partial (f \circ z)(x(t))$ can be written as a convex combination of a finite set of points in the form of $\sum_{i=1}^n \alpha_i h_i$. So $(f \circ z \circ x)'(t) = \langle k, x'(t) \rangle$ holds for a.e. $t > 0$. \qed

H Additional Experimental Results

In this section, we provide additional results of our experiments.

H.1 Loss-based Learning Rate Scheduling

The intuition of the loss-based learning rate scheduling is as follows. If the training loss is $\alpha$-smooth, then optimization theory suggests that we should set the learning rate to roughly $1/\alpha$. For a homogeneous model with cross-entropy loss, if the training accuracy is 100% at $\theta$, then a simple calculation can show that the smoothness (the $L^2$-norm of the Hessian matrix) at $\theta$ is $O(\mathcal{L} \cdot \text{poly}(\rho))$, where $\mathcal{L}$ is the average training loss and $\text{poly}(\rho)$ is some polynomial. Motivated by this fact, we parameterize the learning rate $\eta(t)$ at epoch $t$ as

$$\eta(t) := \frac{\alpha(t)}{\mathcal{L}(t - 1)},$$

where $\alpha(t)$ is some polynomial in $t$.
where $\bar{L}(t - 1)$ is the average training loss at epoch $t - 1$, and $\alpha(t)$ is a relative learning rate to be tuned (Similar parameterization has been considered in [Nacson et al., 2018a] for linear model). The loss-based learning rate scheduling is indeed a variant of line search. In particular, we initialize $\alpha(0)$ by some value, and do the following at each epoch $t$:

Step 1. Initially $\alpha(t) \leftarrow \alpha(t - 1)$; Let $\bar{L}(t - 1)$ be the training loss at the end of the last epoch;

Step 2. Run SGD through the whole training set with learning rate $\eta(t) := \alpha(t)/L(t - 1)$;

Step 3. Evaluate the training loss $\bar{L}(t)$ on the whole training set;

Step 4. If $\bar{L}(t) < L(t - 1)$, $\alpha(t) \leftarrow \alpha(t) \cdot r_u$ and end this epoch; otherwise, $\alpha(t) \leftarrow \alpha(t)/r_d$ and go to Step 2.

In all our experiments, we set $\alpha(0) := 0.1$, $r_u := 2^{1/5} \approx 1.149$, $r_d := 2^{1/10} \approx 1.072$. This specific choice of those hyperparameters is not important; other choices can only affect the computational efficiency, but not the overall tendency of normalized margin.

H.2 Additional Figures

In Figure 3 and 4, we plot curves for the training accuracy and the normalized margin (with the original definition in Section 3.1) of CNNs with and without bias during training.

Figure 3: Training CNNs with and without bias on MNIST, using SGD with learning rate 0.01. The training accuracy (left) increases to 100% after about 100 epochs, and the normalized margin with the original definition (right) keeps increasing after the model is fitted.

H.3 Addressing Numerical Issues

Since we are dealing with extremely small loss (as small as $10^{-800}$), the current Tensorflow implementation would run into numerical issues. To address the issues, we work as follows. Let $\bar{L}_B(\theta)$ be the (average) training loss within a batch $B \subseteq [N]$. We use the notations $C, s_n, q_n, \tilde{q}_n$ from Appendix D. We only need to show how to perform forward and backward passes for $\bar{L}_B(\theta)$.

**Forward Pass.** Suppose we have a good estimate $\tilde{F}$ for $\log \bar{L}_B(\theta)$ in the sense that

$$R_B(\theta) := \bar{L}_B(\theta)e^{-\tilde{F}} = \frac{1}{B} \sum_{n \in B} \log \left( 1 + \sum_{j \neq y_n} e^{-s_n j(\theta)} \right) e^{-\tilde{F}}$$  \hspace{1cm} (19)

is in the range of float64. $R_B(\theta)$ can be thought of a relative training loss with respect to $\tilde{F}$. Instead of evaluating the training loss $\bar{L}_B(\theta)$ directly, we turn to evaluate this relative training loss in a numerically stable way:
Figure 4: Training CNNs with and without bias on MNIST, using SGD with the loss-based learning rate scheduler. The training accuracy (left) increases to 100% after about 20 epochs, and the normalized margin with the original definition (middle) increases rapidly after the model is fitted. The right figure shows the change of the relative learning rate $\alpha(t)$ (see (18) for its definition) during training.

Step 1. Perform forward pass to compute the values of $s_{nj}$ with float32, and convert them into float64;

Step 2. Let $Q := 30$. If $q_n(\theta) > Q$ for all $n \in B$, then we compute

$$R_B(\theta) = \frac{1}{B} \sum_{n \in B} e^{-(\tilde{q}_n(\theta) + \tilde{F})},$$

where $\tilde{q}_n(\theta) := -\text{LSE}\{-s_{nj} : j \neq y_n\} = -\log\left(\sum_{j \neq y_n} e^{-s_{nj}}\right)$ is evaluated in a numerically stable way; otherwise, we compute

$$R_B(\theta) = \frac{1}{B} \sum_{n \in B} \log\text{1p}\left(\sum_{j \neq y_n} e^{-s_{nj}(\theta)}\right) e^{-\tilde{F}},$$

where $\log\text{1p}(x)$ is a numerical stable implementation of $\log(1 + x)$.

This algorithm can be explained as follows. Step 1 is numerically stable because we observe from the experiments that the layer weights and layer outputs grow slowly. Now we consider Step 2. If $q_n(\theta) \leq Q$ for some $n \in [B]$, then $\mathcal{L}_B(\theta) = \Omega(e^{-Q})$ is in the range of float64, so we can compute $R_B(\theta)$ by (19) directly except that we need to use a numerical stable implementation of $\log(1 + x)$. For $q_n(\theta) > Q$, arithmetic underflow can occur. By Taylor expansion of $\log(1 + x)$, we know that when $x$ is small enough $\log(1 + x) \approx x$ in the sense that the relative error $\frac{\log(1 + x) - x}{\log(1 + x)} = O(x)$. Thus, we can do the following approximation

$$\log\left(1 + \sum_{j \neq y_n} e^{-s_{nj}(\theta)}\right) e^{-\tilde{F}} \approx \sum_{j \neq y_n} e^{-s_{nj}(\theta)} e^{-\tilde{F}},$$

(20)

for $q_n(\theta) > Q$, and only introduce a relative error of $O(C e^{-Q})$ (recall that $C$ is the number of classes). Using a numerical stable implementation of LSE, we can compute $\tilde{q}_n$ easily. Then the RHS of (20) can be rewritten as $e^{-(\tilde{q}_n(\theta) + \tilde{F})}$. Note that computing $e^{-(\tilde{q}_n(\theta) + \tilde{F})}$ does not have underflow or overflow problems if $\tilde{F}$ is a good approximation for $\log \mathcal{L}_B(\theta)$.

**Backward Pass.** To perform backward pass, we build a computation graph in Tensorflow for the above forward pass for the relative training loss and use the automatic differentiation. We parameterize the learning rate as $\eta = \tilde{\eta} \cdot e^F$. Then it is easy to see that taking a step of gradient descent for $\mathcal{L}_B(\theta)$ with learning rate $\eta$ is equivalent to taking a step for $R_B(\theta)$ with $\tilde{\eta}$. Thus, as long as $\tilde{\eta}$ can fit into float64, we can perform gradient descent on $R_B(\theta)$ to ensure numerical stability.
The Choice of $\tilde{F}$. The only question remains is how to choose $\tilde{F}$. In our experiments, we set $\tilde{F}(t) := \log \bar{L}(t-1)$ to be the training loss at the end of the last epoch, since the training loss cannot change a lot within one single epoch. For this, we need to maintain $\log \bar{L}(t)$ during training. This can be done as follows: after evaluating the relative training loss $\mathcal{R}(t)$ on the whole training set, we can obtain $\log \bar{L}(t)$ by adding $\tilde{F}(t)$ and $\log \mathcal{R}(t)$ together.

It is worth noting that with this choice of $\tilde{F}$, $\hat{\eta}(t) = \alpha(t)$ in the loss-based learning rate scheduling. As shown in the right figure of Figure 4, $\alpha(t)$ is always between $10^{-9}$ and $10^{0}$, which ensures the numerical stability of backward pass.

H.4 Experiments on CIFAR-10

To verify whether the normalized margin is increasing in practice, we also conduct experiments on CIFAR-10. We use a modified version of VGGNet-16. The layers of this model can be described as $\text{conv}-64 \times 2, \text{max-pool}, \text{conv}-128 \times 2, \text{max-pool}, \text{conv}-256 \times 3, \text{max-pool}, \text{conv}-512 \times 3, \text{max-pool}, \text{conv}-512 \times 3, \text{max-pool}, \text{fc}-10$ in order, where each conv has filter size $3 \times 3$. We train two networks: one is exactly the same as the VGGNet we described, and the other one is the VGGNet without any bias terms except those in the first layer (similar as in the experiments on MNIST).

The experiment results are shown in Figure 5 and 6. We can see that the normalize margin is increasing over time.

![Figure 5: Training VGGNet with and without bias on CIFAR-10, using SGD with learning rate 0.1.](image)

![Figure 6: Training VGGNet with and without bias on CIFAR-10, using SGD with the loss-based learning rate scheduler.](image)