Convergence of Gradient Descent on Separable Data

Mor Shpigel Nacson[1] **Jason D. Lee**[2] **Suriya Gunasekar**[3] **Pedro H. P. Savarese**[3] **Nathan Srebro**[3] **Daniel Soudry**[1]
[1]Technion, Israel, [2]USC Los Angeles, USA, [3]TTI Chicago, USA

Abstract

We provide a detailed study on the implicit bias of gradient descent when optimizing loss functions with strictly monotone tails, such as the logistic loss, over separable datasets. We look at two basic questions: (a) what are the conditions on the tail of the loss function under which gradient descent converges in the direction of the L_2 maximum-margin separator? (b) how does the rate of margin convergence depend on the tail of the loss function and the choice of the step size? We show that for a large family of super-polynomial tailed losses, gradient descent iterates on linear networks of any depth converge in the direction of L_2 maximum-margin solution, while this does not hold for losses with heavier tails. Within this family, for simple linear models we show that the optimal rates with fixed step size is indeed obtained for the commonly used exponentially tailed losses such as logistic loss. However, with a fixed step size the optimal convergence rate is extremely slow as $1/\log(t)$, as also proved in Soudry et al. (2018a). For linear models with exponential loss, we further prove that the convergence rate could be improved to $\log(t)/\sqrt{t}$ by using aggressive step sizes that compensates for the rapidly vanishing gradients. Numerical results suggest this method might be useful for deep networks.

1 INTRODUCTION

In learning over-parameterized models, where the training objective has multiple global optima, each optimization algorithm can have a distinct implicit bias. Hence, different algorithms learn different models with different generalization to the population loss. This effect of the implicit bias of the optimization algorithm on the learned model is particularly prominent in deep learning, where the generalization or the inductive bias is not sufficiently driven by explicit regularization or restrictions on the model capacity (Neyshabur et al., 2015; Zhang et al., 2017; Hoffer et al., 2017). Thus, in order to understand what is the true inductive bias in such high capacity models, it is important to rigorously understand how optimization affects the implicit bias.

Consider learning a homogeneous linear predictor $\mathbf{x} \to \mathbf{w}^\top \mathbf{x}$ using unregularized logistic regression over separable data. For this problem, Soudry et al. (2018a) showed that the gradient descent iterates converge in direction to the maximum-margin separator with unit L_2 norm, and this implicit bias holds independently of initialization and step size (given the step size is small enough). This is exactly the solution of the homogeneous hard margin support vector machine (SVM) where the L_2 norm constraint on the parameters $\mathbf{w}$ is explicitly added. More surprisingly, Soudry et al. (2018a) also showed that the rate of convergence to the maximum-margin solution is $O(1/\log(t))$. This is much slower compared to the rate of convergence of the loss function itself, which is shown to be $O(1/t)$. This implies that the classification boundary of logistic regression, and hence the generalization of the classifier, continues to change long after the 0-1 error on training examples has diminished to zero, or the logistic loss is very small. In a follow up work, Gunasekar et al. (2018a) showed that for exponential loss, gradient descent on fully connected deep linear networks also has the same bias asymptotically. However, the convergence rates were not analyzed in this work on deep linear networks.

Despite this recent line of interesting results, the implicit bias of gradient descent is not entirely understood even in simple linear classification tasks. For example, the analysis of Soudry et al. (2018a) and Gunasekar et al. (2018a) crucially relied on strict monotonicity of the loss function to get an initialization–independent characterization of the bias of gradient descent. However, in these work the results are derived specifically for tight exponential tailed losses and exponential loss, respectively. While exponential tailed losses such as logistic and cross entropy losses are indeed the most widely used losses in training deep neural net-

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

works, we do not yet know: Do such losses with tight exponential tail have a special significance? Can a similar convergence to maximum-margin separator be achieved by other strictly monotonic losses? How is the rate of convergence to maximum-margin solution affected by the tail? Are there other ways to accelerate the convergence?

Here we provide a detailed study of this problem, focusing on the rate of convergence of the margin:

1. *What are the conditions on the tail of the loss function under which gradient descent converges to the L_2 maximum-margin separator?* We show that convergence to the L_2 maximum-margin solution can be extended to losses with super polynomial tails, but not to losses with (sub) polynomial tails.
2. *Does a heavier or lighter tail gives a faster rate of convergence?* In our analysis, losses with exponential tails, which include the commonly used logistic loss, can indeed be shown to have the optimal rate of convergence of the margin.
3. *Extensions to deep linear networks.* We show that similar analysis and the same asymptotic rates hold more generally for linear networks with fully connected layers. Interestingly, the results suggest that increasing the number of layers (depth) decreases the convergence rate only marginally, even in the limit of infinite depth.
4. *For exponential loss, which obtains the optimal margin convergence rate, can we accelerate the convergence to the maximum-margin by using variable step sizes?* The answer is yes, and we show that using normalized gradient updates, i.e., step size proportional to the inverse gradient, we can get a much faster rate of $O(\log t/\sqrt{t})$ instead of $1/\log t$. Experimental results suggests this improvement in rate over standard gradient descent might also extend for non-linear neural networks.

2 SETUP AND REVIEW OF PREVIOUS RESULTS

Consider a dataset $\{\mathbf{x}_n, y_n\}_{n=1}^N$, with features $\mathbf{x}_n \in \mathbb{R}^d$ and binary labels $y_n \in \{-1, 1\}$. All the results in the paper are stated for data $\{\mathbf{x}_n, y_n\}_{n=1}^N$ which is *strictly linearly separable*, i.e., there exists a separator $\mathbf{w}_*$ such that $\forall n : y_n \mathbf{w}_*^\top \mathbf{x}_n > 0$.

We study learning homogenous linear predictors by minimizing unregularized empirical losses of the form

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \ell\left(y_n \mathbf{w}^\top \mathbf{x}_n\right) , \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector or the linear predictor. To simplify notation, we assume that $\forall n : y_n = 1$ — this is without loss of generality, since we can always re-define $y_n \mathbf{x}_n$ as $\mathbf{x}_n$. We denote the data matrix by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ and $\|\cdot\|$ denotes the L_2 norm.

The gradient descent (GD) updates for minimizing $\mathcal{L}(\mathbf{w})$ in eq. (1) with step size sequence $\{\eta_t\}_{t=0}^\infty$ is given by

$$\begin{aligned} \mathbf{w}(t+1) &= \mathbf{w}(t) - \eta_t \nabla \mathcal{L}(\mathbf{w}(t)) \\ &= \mathbf{w}(t) - \eta_t \sum_{n=1}^{N} \ell'(\mathbf{w}(t)^\top \mathbf{x}_n) \mathbf{x}_n . \end{aligned} \tag{2}$$

We look at the iterates of GD on linearly separable datasets with monotonic loss functions.

Definition 1. *[Strict Monotone Loss] $\ell(u)$ is a differentiable strictly monotonically decreasing function bounded from below, i.e. $\forall u, \ell(u)' < 0$ and, without loss of generality, $\forall u, \ \ell(u) > 0$ and $\lim_{u\to\infty} \ell(u) = \lim_{u\to\infty} \ell'(u) = 0$. Also, $\limsup_{u\to-\infty} \ell'(u) \neq 0$.*

Examples of strict monotone losses include common classification losses such as logistic loss, exponential loss, and probit loss. A key property of interest with such losses is that the empirical risk in eq. (1) over separable data does not have any finite global minimizers. Thus, whenever the gradient descent updates in eq. (2) minimize the empirical loss $\mathcal{L}(\mathbf{w})$, the iterates $\mathbf{w}(t)$ will necessarily diverge to infinity. Nevertheless, in this case, even though the norm of the iterates $\|\mathbf{w}(t)\|$ diverge, the classification boundary is entirely specified by the direction of $\mathbf{w}(t)/\|\mathbf{w}(t)\|$. Can we say something interesting about which direction the iterates $\mathbf{w}(t)$ converge to?

For monotone losses with $-\ell'(u)$ satisfying the specific *tight exponential tail* property (defined below), Soudry et al. (2018a) characterized this direction to be the maximum-margin separator,

Definition 2. *[Tight Exponential Tail] A scalar function $h(u)$ has a tight exponential tail, if there exist positive constants μ_+, μ_-, and $\bar{u}$ such that $\forall u > \bar{u}$:*

$$(1 - \exp(-\mu_- u))e^{-u} \leq h(u) \leq (1 + \exp(-\mu_+ u))e^{-u}.$$

Theorem 1 (Theorem 3 in Soudry et al. (2018a), rephrased)**.** *For almost all linearly separable datasets $\{\mathbf{x}_n, y_n\}_{n=1}^N$, and any β-smooth $\mathcal{L}$ with a strictly monotone loss function ℓ (Definition 1), for which $-\ell'$ has a tight exponential tail (Definition 2), the gradient descent iterates $\mathbf{w}(t)$ in eq. (2) with any fixed step size satisfying[1] $\eta < 2\beta^{-1}$ and any initialization $\mathbf{w}(0)$, will behave as:*

$$\mathbf{w}(t) = \hat{\mathbf{w}} \log t + \boldsymbol{\rho}(t) , \tag{3}$$

[1]Note that for exponential loss $\ell(u) = \exp(-u)$, $\mathcal{L}(\mathbf{w})$ does not have a global smoothness parameter β. However, with $\eta < 1/\mathcal{L}(\mathbf{w}(0))$ it is straightforward to show the gradient descent iterates maintain bounded local smoothness $\beta(t) \leq \mathcal{L}(\mathbf{w}(t)) \leq \mathcal{L}(\mathbf{w}(0))$, so we will have $\eta < \beta(t)^{-1}$ for all iterates, which suffices for the result to extend to exponential loss.

where the residual $\boldsymbol{\rho}(t)$ is bounded and $\hat{\mathbf{w}}$ is the following L_2 max margin separator:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \|\mathbf{w}\|^2 \text{ s.t. } y_n \mathbf{w}^\top \mathbf{x}_n \geq 1. \tag{4}$$

In Theorem 1 and in the remainder of the paper, *almost all* datasets refers to all datasets except a measure zero set of $\{\mathbf{x}_n\}_n$, e.g., with probability 1, any dataset sampled from an absolutely continuous distribution.

Interestingly, and somewhat surprisingly, Theorem 1 implies logarithmically slow convergence in direction to the L_2 max-margin separator. This slow convergence rate also applies to the margin. This, in contrast, is much slower compared to the rate of convergence of the loss $\mathcal{L}(\mathbf{w}(t))$ itself, which can be shown to decay as $O(1/t)$ (see Lemma 1 in Soudry et al. (2018b)).

Multilayer linear networks. In a recent follow up work, Gunasekar et al. (2018a) extend such results to fully connected deep linear networks, where the objective is non-convex. A multi-layer linear network consists of nodes arranged in L layers. We use the convention that for an L layer network, the inputs features $\mathbf{x}$ form the source nodes in the zeroth layer $l = 0$ and the output is sink node in the final layer $l = L$. Let d_l for $l = 0, 1, \ldots, L$ denote the number of nodes in layer l. The network is parameterized by weight matrices $\mathcal{W} = \{\mathbf{W}_l \in \mathbb{R}^{d_{l-1}\times d_l} : l = 1, 2, \ldots, L\}$. Every such network represents a linear mapping given as follows:

$$\mathbf{w} = \mathcal{P}(\mathcal{W}) := \mathbf{W}_1 \cdot \mathbf{W}_2 \cdot \ldots \cdot \mathbf{W}_L \in \mathbb{R}^d.$$

Unlike logistic regression, where the parameters of the linear model $\mathbf{w} \in \mathbb{R}^d$ are learned directly by minimizing the training loss, in training linear networks, the objective is instead minimized over the parameters of the network $\mathcal{W} = \{\mathbf{W}_l \in \mathbb{R}^{d_{l-1}\times d_l} : l = 1, 2, \ldots, L\}$. The empirical loss is given by:

$$\mathcal{L}_{\mathcal{P}}(\mathcal{W}) = \mathcal{L}(\mathcal{P}(\mathcal{W})) = \sum_{n=1}^{N} \ell\left(y_n \left\langle \mathcal{P}(\mathcal{W}), \mathbf{x}_n \right\rangle\right). \tag{5}$$

Gradient descent iterates $\mathcal{W}(t) = \{\mathbf{W}_l(t)\}_{l=1}^L$ for the above objective are given by:

$$\forall l, \mathbf{W}_l(t+1) = \mathbf{W}_l(t) - \eta_t \nabla_{\mathbf{W}_l} \mathcal{L}_{\mathcal{P}}(\mathcal{W}(t)), \tag{6}$$

and the corresponding sequence of linear predictors along the gradient descent path is given by,

$$\mathbf{w}(t) = \mathcal{P}(\mathcal{W}(t)) = \mathbf{W}_1(t) \cdot \ldots \cdot \mathbf{W}_L(t) \in \mathbb{R}^d. \tag{7}$$

For the special case of exponential loss, Gunasekar et al. (2018a) showed that the linear separator $\mathbf{w}(t)$ in eq. (6) learned by gradient descent on fully connected network (under additional conditions on convergence of the net parameters and gradients, and convergence of the loss) again converges in the direction of the L_2 maximum-margin separator (Theorem 1 in Gunasekar et al. (2018a)). This result, however, only applies to exponential loss and does not specify how quickly the margin of $\mathbf{w}(t)$ converges to the maximum-margin (in case of convergence).

3 MAIN RESULTS

In this section, we provide a detailed analysis of the implicit bias in linear models focusing on convergence and rate of convergence of margin under general tails and with variable step sizes. We use the following standard notation on asymptotic behaviour: (a) $f(u) = \omega(g(u)) \Leftrightarrow \lim_{u\to\infty} \left|\frac{f(u)}{g(u)}\right| = \infty$, (b) $f(u) = o(g(u)) \Leftrightarrow \lim_{u\to\infty} \frac{f(u)}{g(u)} = 0$, (c) $f(u) = O(g(u)) \Leftrightarrow \limsup_{u\to\infty} \frac{|f(u)|}{g(u)} < \infty$, (d) $f(u) = \Omega(g(u)) \Leftrightarrow \lim\inf_{u\to\infty} \frac{f(u)}{g(u)} > 0$, and (e) $f(u) = \Theta(g(u)) \Leftrightarrow \Omega(g(u)) = f(u) = O(g(u))$.

Previous results, summarized in Section 2, show that when minimizing exponentially tailed losses on separable datasets, gradient descent converges to the L_2 max-margin separator with a very slow rate of $1/\log(t)$. While commonly used classification losses such as logistic loss, cross entropy loss, and exponential loss indeed have tight exponential tail, the significance of the exponential tail is not fully understood. What are the general conditions on the tail under which gradient descent converges to the maximum-margin solution? Can the rate of convergence be accelerated by choosing a heavier or lighter tail?

3.1 Linear networks with general tails

We first show that for a large family of strictly monotone losses with super–polynomial tails specified (Assumption 1 below), gradient descent iterates converge to the maximum-margin solution. We will later also analyze the rate of convergence for this family of loss functions.

Assumption 1. *$\ell(u)$ is analytic and satisfies the following:*

1. ***Strict monotonicity:*** *ℓ satisfies Definition 1. Since, $\forall u,\ \ell'(u) < 0$, let $\ell'(u) = -\exp(-f(u))$.*
2. ***Super-polynomial tail:*** *$\ell(u)$ has a "super-polynomial tail" if $\forall M > 0$, $\exists u_0$ such that $\forall u \geq u_0$, $-\ell'(u) \leq u^{-M}$. This is equivalent to $f(u) = \omega(\log(u))$.*
3. ***Asymptotically convex:*** *$\exists u_0$ such that $\forall u > u_0$, $\ell''(u) > 0$. For strictly monotone decreasing losses, this is equivalent to $\forall u > u_0$, $f'(u) = \frac{\ell''(u)}{-\ell'(u)} > 0$.*
4. ***Non-oscillatory tail:*** *$\lim_{u\to\infty} uf'(u)$ exists. For losses with super-polynomial tails where $f(u) = \omega(\log(u))$, this condition implies $f'(u) = \omega\left(u^{-1}\right)$.*

Remark 1. *Assumption 1 captures a large family strictly monotone losses with super-polynomial tails that are relevant for binary classification tasks, and the last condition is rather technical to avoid undesirable oscillatory behaviour*

like $f(u) = u + \sin(u)$. In particular, the assumption includes the following special cases:

- *Logistic loss $\ell(u) = \log(1 + e^{-u})$, for which $f(u) = \log(1 + e^u) = \omega(\log(u))$ and $f'(u) = \frac{e^u}{1+e^u} = \omega\left(u^{-1}\right)$.*
- *Other losses with tight exponential tail (Definition 2), like the exponential loss $\ell(u) = \exp(-u)$.*
- *"Poly-exponential" tailed losses given by $\ell'(u) = -\exp(-u^\nu)$ for degree $\nu > 0$, e.g., the probit loss.*
- *Sub-exponential super-polynomial tails like $\ell'(u) = -u^{-\log^\mu(u)}$ for $\mu > 0$.*

For depth–L linear networks, we first show that the implicit bias of gradient descent for exponential loss from Gunasekar et al. (2018a) can be extended more broadly to super-polynomial tailed losses specified in Assumption 1.

Theorem 2. *For any depth L, almost all linearly separable datasets, almost all initialization and any bounded sequence of step sizes $\{\eta_t\}$, consider the sequence $\mathcal{W}(t) = \{\mathbf{W}_l(t)\}_{l=1}^L$ of gradient descent updates in eq. (6) for minimizing the empirical loss $\mathcal{L}_\mathcal{P}(\mathcal{W})$ (eq. (5)) with a strictly monotone loss function ℓ satisfying Assumption 1, i.e.: $\ell'(u) = -\exp(-f(u)) < 0$, where asymptotically $f'(u) > 0$ and $f'(u) = \omega\left(u^{-1}\right)$.*

If (a) $\mathcal{W}(t)$ minimizes the empirical loss, i.e. $\mathcal{L}_\mathcal{P}\left(\mathcal{W}(t)\right) \to 0$, (b) $\mathcal{W}(t)$, and consequently $\mathbf{w}(t) = \mathcal{P}\left(\mathbf{w}(t)\right)$, converge in direction to yield a separator with positive margin, and (c) the gradients with respect to the linear predictors $\nabla_\mathbf{w}\mathcal{L}\left(\mathbf{w}(t)\right)$ converge in direction, then the limit direction is given by,

$$\bar{\mathbf{w}}_\infty = \lim_{t\to\infty} \frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|},$$

where

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}\|^2 \text{ s.t. } \mathbf{w}^\top\mathbf{x}_n \geq 1. \tag{8}$$

This theorem is proved in Appendix D, while the basic ideas are sketched in Appendix C for $L = 1$.

Remark 2. *Theorem 2 covers a large family of super-polynomial tails specified under Assumption 1. Conversely, for (sub) polynomial tails, we may not converge to the maximum-margin separator. In Appendix H, we show that we do not converge to the max margin if $\ell(u)$ has polynomial tail. Additionally, with the hinge loss (which it is neither differentiable or strictly monotonic) we generally do not converge to the maximum-margin without regularization, as then GD typically converges to a finite minimizer that depends on the initialization.*

Remark 3. *Rosset et al. (2004) also investigated the connection between the loss function choice and the maximum-margin solution. In this work, Rosset et al. (2004) considered linear models with monotone loss functions and explicit norm regularization. We discuss the connections between Rosset et al. (2004) results and ours in appendix A.*

Remark 4. *Gunasekar et al. (2018a) characterized the implicit bias of gradient descent for fully connected linear networks for the special case of exponential loss $\ell(u) = \exp(-u)$. Theorem 2 generalizes this characterization to a larger family of losses, which in particular includes the commonly used logistic loss. Logistic loss, despite having the same exponential tail as the exponential loss, was not explicitly analyzed in Gunasekar et al. (2018a).*

We now continue to characterizing the convergence rates.

3.2 Rates of convergence

To calculate the convergence rates we will make an additional assumption.

Assumption 2. *$f(u)$ is real analytic on $\mathbb{R}_{++}$ and satisfies $\forall k \in \mathbb{N} : \left|\frac{f^{(k+1)}(u)}{f'(u)}\right| = O\left(u^{-k}\right)$.*

While the above assumption is not required to show asymptotic convergence of gradient descent to the maximum-margin separator (Theorem 2), we do require the additional assumption to calculate the rates. This assumption implies that the loss tail does not decay too fast. In particular, Assumption 2 is *not* satisfied by super-poly-exponential tails like $\ell'(u) = \exp\left(-\exp(u^\nu)\right)$ for $\nu > 0$ or $\ell'(u) = \exp\left(-\exp\left(\log^\mu(u)\right)\right)$ for $\mu > 1$, and additionally avoids oscillatory functions like $\sin(u)$.

Nevertheless, a large class of interesting monotone functions satisfy this assumption, including cases where $f(u)$ is polynomial and poly-logarithmic functions. Within this family, we look at the margin rate of convergence of the gradient descent iterates, for $L = 1$ in two regimes:

1. $f'(u) = \omega(1)$, which implies $-\ell'(u) = \omega(\exp(-u))$. This case includes loss functions with tails *lighter* than the exponential tail, for example poly-exponential tail $\ell(u) = \exp(-u^\nu)$ with s strictly greater than one exponent, $\nu > 1$.
2. $f'(u) = \omega(u^{-1})$ and $f'(u) = o(1)$: or $-\ell'(u) = o(\exp(-u))$. This case includes loss functions with tails *heavier* than the exponential tail, such as $\ell(u) = \exp(-u^\nu)$ for $\nu < 1$ or $\ell(u) = \exp(-\log^\mu(u))$ for $\mu > 0$.

We first look at the rates for the special case of $L = 1$ where the parameters $\mathbf{w}$ of the linear models are directly learned using gradient descent. This is the setting analyzed in Soudry et al. (2018a) with tight exponential tailed losses. The following theorem is proved in Appendix F.

Theorem 3. *For almost all linearly separable datasets, almost all initialization, any bounded sequence of step sizes $\{\eta_t\} < 2\beta^{-1}$, and a single layer $L = 1$, consider the sequence of gradient descent updates in eq. (2) for minimizing the empirical loss $\mathcal{L}(\mathbf{w})$ (eq. (1)) with a strictly monotone β-smooth loss function ℓ satisfying*

$\ell'(u) = -\exp(-f(u)) < 0$, *where asymptotically* $f'(u) = \Omega\left(\frac{1}{u}\log^{1+\epsilon}(u)\right)$ *for some* $\epsilon > 0$ *and satisfies Assumption 2.*

If (a) $\mathbf{w}(t)$ *converges in direction to yield a separator with positive margin, and (b) the gradients with respect to the linear predictors* $\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}(t))$ *converge in direction, then the margin convergence of* $\mathbf{w}(t)$ *to the max margin* $\gamma = \max_{\mathbf{w}}\min_n \frac{\mathbf{w}^\top \mathbf{x}_n}{\|\mathbf{w}\|}$ *satisfies:*

1. *If* $f'(u) = \omega(1)$ *(which implies* $f(t) = \omega(t)$*), then*

$$\gamma - \min_n \frac{\mathbf{x}_n^\top \mathbf{w}(t)}{\|\mathbf{w}(t)\|} = O\left(\frac{1}{f^{-1}(\log(t))}\right).$$

2. *If* $f'(u) = o(1)$ *and* f *is strictly concave, then*

$$\gamma - \min_n \frac{\mathbf{x}_n^\top \mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \Omega\left(\frac{1}{\log(t)}\right)$$

and the optimal rate is obtained for exponential loss.

From the proof of Theorem 3, we can also calculate the rates of convergence for the normalized direction $\mathbf{w}(t)/\|\mathbf{w}(t)\|$ to the maximum-margin separator $\hat{\mathbf{w}}/\|\hat{\mathbf{w}}\|$, as well as the convergence of the angle between them.

Corollary 1. *We examine super-polynomial tailed losses satisfying the assumptions of the previous Theorem, when the loss tail does not decay too fast, i.e.* $\left|\frac{f'(u)}{f(u)}\right| = O(u^{-1})$. *The optimal rate of convergence to the max margin of GD with fixed step size is* $1/\log(t)$. *This optimal rate is attained by exponentially tailed losses, where* $f(u) = \Theta(u)$ *(or* $f'(u) = \Theta(1)$*). This includes the popular losses of logistic loss and exponential loss.*

Proof. For the case of $f'(u) = \omega(1)$, $f(t) = \omega(t) \Rightarrow f^{-1}(t) = o(t)$ and thus, the rate for this case $O\left(\frac{1}{f^{-1}(\log(t))}\right)$ is sub-optimal compared with the rate for exponential loss which is $1/\log(t)$ (from Theorem 1). In appendix sections H.3, H.4 we give a positive example that demonstrates that this upper bound is tight, i.e., it is obtained for some datasets, and a negative example which shows a case in which the upper bound is not obtained. In general as long as the loss tail does not decay too fast, i.e. $\left|\frac{f'(u)}{f(u)}\right| = O(u^{-1})$, the rate in this case is $\Omega\left(\frac{1}{\log(t)}\right)$ (see appendix E.5). Secondly, for the case of $f'(u) = o(1)$ the asymptotic rate is $\Omega(1/\log(t))$, so the optimal rate we can hope for with any tail is $O(1/\log(t))$. In appendix F we show that the exponential tail obtains this optimal rate. Additionally, in Appendix J, we show that for the special case of poly-exponential losses $\ell'(u) = -\exp(-u^\nu)$ with $0.25 < \nu \leq 1$, the rate is indeed $O(1/\log(t))$ and the constants in the rates for $\nu < 1$ are strictly worse than that of exponential tail with $\nu = 1$. □

Remark 5. *Note that for* $L = 1$ *the optimization objective (eq.* (5) *) is convex in the optimization variables and hence, by Lemma* 1 *in Soudry et al. (2018a), the assumption in Theorem 2 that* $\mathcal{L}_P(\mathcal{W}(t)) \to 0$ *is satisfied for appropriate choices of step size. Moreover for the special case of poly-exponential tails with* $\ell'(u) = -\exp(-u^\nu)$ *for* $\nu > 0.25$*, the convergence to the maximum-margin separator and the convergence rates can be obtained without the assumptions that* $\mathbf{w}(t)$ *and* $\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}(t))$ *converge in direction (see Appendix J).*

Now we state the results for the general case of L–layer linear network.

Theorem 4. *Under assumption 2 and the conditions and notations of Theorem 2, if the SVM support vectors span the data then for any depth* L *the network equivalent linear predictor* $\mathbf{w}(t)$ *satisfies:*

$$\gamma - \min_n \frac{\mathbf{x}_n^\top \mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \begin{cases} O\left(\frac{1}{g(t)}\right), & f'(u) = \omega(1) \\ \Theta\left(\frac{1}{g(t)f'(g(t))}\right), & \text{otherwise} \end{cases}$$

where $g(t)$ *is the asymptotic solution of*

$$\frac{dg(t)}{dt} = -\ell'(g(t))\left(g(t)\right)^{2\left(1-L^{-1}\right)}. \tag{9}$$

Remark 6. *Importantly, from Assumption 1,* $-\ell'(u)$ *has super-polynomial tail, which suggests the factor* $\left(g(t)\right)^{2\left(1-L^{-1}\right)}$ *only negligibly affects the asymptotic solution of eq.* (9)*. This implies that* $\forall L > 1$*, and even in the limit* $L \to \infty$*, the rate predicted by this Theorem 4 will only be slightly smaller than the* $L = 1$ *case of Theorem 3. This difference will become negligible in the limit* $t \to \infty$*. For example, for the case of exponential loss, we prove in appendix E.4 that the ODE solution is* $g(t) = \log(t) + o(\log(t))$*. Thus, in this case, the margin converges as* $O(1/\log(t))$ *for any depth.*

3.3 Faster rates using variable step sizes

Our analysis so far suggests that exponential tails have an optimal convergence rate, and for exponential tail losses with a bounded step size, we have an extremely slow rate of convergence, $O(1/\log t)$. Therefore, the question is can we somehow accelerate this rate using variable unbounded step sizes. Fortunately, at least for linear models trained with exponential loss, the answer is yes and we can indeed show faster rate of convergence by aggressively increasing the step size to compensate for the vanishing gradient. Specially, we examine the following normalized GD algorithm:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \frac{\nabla\mathcal{L}(\mathbf{w}(t))}{\|\nabla\mathcal{L}(\mathbf{w}(t))\|}. \tag{10}$$

Recall that $\gamma = \max_{\mathbf{w}:\|\mathbf{w}\|\leq 1}\min_n \mathbf{w}^\top \mathbf{x}_n$ is the maximum-margin of the dataset with unit L_2 norm separators, and without loss of generality assume $\forall n : \|\mathbf{x}_n\| \leq 1$.

By the triangle inequality, we have that $\|\nabla\mathcal{L}(\mathbf{w}(t))\| = \left\|\sum_n \exp(-\mathbf{w}(t)^\top \mathbf{x}_n)\mathbf{x}_n\right\| \leq \mathcal{L}(\mathbf{w}(t))$. We additionally have the following inequality for all t,

$$\|\nabla\mathcal{L}(\mathbf{w}(t))\| = \max_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \sum_n \exp(-\mathbf{w}(t)^\top \mathbf{x}_n)\mathbf{w}^\top \mathbf{x}_n$$
$$\geq \gamma \sum_n \exp(-\mathbf{w}(t)^\top \mathbf{x}_n) = \gamma\mathcal{L}(\mathbf{w}(t))\,.$$

Thus, for all $\mathbf{w}$, the two-sided bound

$$\gamma\mathcal{L}(\mathbf{w}) \leq \|\nabla\mathcal{L}(\mathbf{w})\| \leq \mathcal{L}(\mathbf{w})$$

holds, and, up to a scaling of step–sizes, the normalized GD in eq. (10) can be alternatively expressed as the following

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \frac{\nabla\mathcal{L}(\mathbf{w}(t))}{\mathcal{L}(\mathbf{w}(t))}\,. \tag{11}$$

We chose to state our results in terms of eq. (11) (normalizing GD by $\mathcal{L}(\mathbf{w}(t))$) so that the stepsize choice η_t does not depend on the optimal margin γ which is unknown. The following theorem proved in Appendix B.1 shows that using normalized GD can improve the rate of convergence of the margin of the separator to $\log t/\sqrt{t}$ compared to $O(1/\log t)$ for fixed step sizes.

Theorem 5. *For any separable data set and any initial point* $\mathbf{w}(0)$*, consider the normalized GD updates in eq.* (11) *with a variable step size* $\eta_t = \frac{1}{\sqrt{t+1}}$ *and exponential loss* $\ell(u) = \exp(-u)$*.*

Then the margin of the iterates $\mathbf{w}(t)$ *converges to the max-margin* γ *with rate* $t^{-1/2}\log t$*:*

$$\frac{\mathbf{w}(t+1)^T\mathbf{x}_n}{\|\mathbf{w}(t+1)\|} \geq \gamma - \frac{1+\log(t+1)}{\gamma(4\sqrt{t+2}-4)} - \frac{\log\mathcal{L}(\mathbf{w}(0))}{\gamma(2\sqrt{t+2}-2)}.$$

In the appendix we prove a more general version of Theorem 5, which obtains the same rate for any steepest descent algorithm. Also, note that normalized GD as in eq. (10) was analyzed before, but for other purposes. For example, Levy (2016) showed a stochastic version of it can better escape saddle points. Here we study the effect of normalization on the implicit bias of the algorithm.

The observation that aggressive changes in the step size can improve convergence rate is applied in the AdaBoost literature (Schapire and Freund, 2012), where exact line-search is used. We use a slightly less aggressive strategy of decaying step sizes with normalized gradient descent, attaining a rate of $\log(t)/\sqrt(t)$. This rate almost matches $1/\sqrt(t)$, which is the optimal rate in terms of margin suboptimality for solving hard margin SVM. This rate is achieved by the best known methods.[2] This suggests that gradient descent with a more aggressive step size policy is quite efficient at margin maximization.

We emphasize our goal here is not to develop a faster SVM optimizer, but rather to understand and improve gradient descent and local search in a way that might be applicable also for deep neural networks, as indicated by the numerical results we present next.

[2]The best known method in terms of margin suboptimality, and using vector operations (operations on all training examples), is the aggressive Perceptron, which achieves a rate of $\sqrt{N/t}$. Clarkson et al. (2012) obtained an improved method which they showed is optimal, that does not use vector operations. Clarkson et al. (2012) method achieves a rate of $\sqrt{(N+d)/t}$, where now t is the number of scalar operations.

4 EXPERIMENTS WITH NORMALIZED GRADIENT DESCENT

In the following experiments, we implement the normalized GD in eq. (10) with step sizes separately tuned for each experiment.

4.1 Linear Networks on Synthetic Data

First, in Figure 1 we visualize the different rates for GD and normalized GD when training a plain logistic regression model on synthetic data. As expected from Theorem 5, we find that normalized GD converges significantly faster than unnormalized GD.

Additionally, we evaluate experimentally the convergence rates of GD and normalized GD for multi-layer linear models. Networks with $L \in \{1, 2, 3\}$ layers and 10 neurons per hidden layer are trained with GD and normalized GD on a synthetic binary classification dataset composed of 600 points, sampled from two normal distributions (one for each class).

We use a fixed learning rate $\eta = 5 \times 10^{-3}$ chosen through grid-search, and train each network for 5×10^4 total iterations. Figure 2 shows the margin gaps during training, with normalized GD providing faster convergence rates across models. Appendix I.2 provides details on data generation and training, along with results on ReLU networks.

4.2 Image Classification on MNIST

The MNIST dataset is composed of 70,000 grayscale images of 0-9 digits (10 classes total), each having 28×28 pixels. We use 10,000 images for testing and the rest for training and validation. Unlike harder datasets such as CIFAR-10 and CIFAR-100, MNIST provides a task where simple models can successfully separate the training examples. Hence, we train a 2-layer feedforward network with 5,000 hidden neurons and ReLU activations ($\text{ReLU}(x) = \max(0, x)$) with full-batch GD and normalized GD using the cross-entropy loss, for a total of 3,000 iterations. We decay the learning rate by a factor of 5 at $50\%, 75\%$ and 87.25% of the total number of iterations.

We performed grid-search over initial learning rate values $\{0.1, 0.3, 0.5, 1.0, 2.5, 5.0\}$ using 5,000 images randomly chosen from the training set as validation, and $\eta = 1.0$

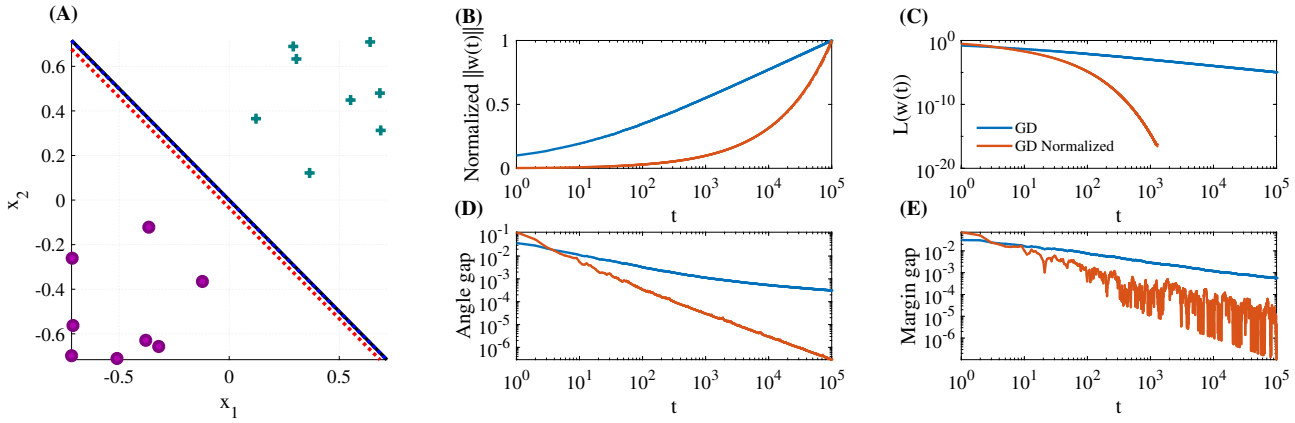

Figure 1: Visualization of the convergence of GD in comparison to normalized GD in a synthetic logistic regression dataset in which the L_2 max margin vector $\hat{\mathbf{w}}$ is precisely known. **(A)** The dataset (positive and negatives samples ($y = \pm 1$) are respectively denoted by $'+'$ and $'\circ'$), max margin separating hyperplane (black line), and the solution of GD (dashed red) and normalized GD (dashed blue) after 10^5 iterations. For both GD and Normalized GD, we show: **(B)** The norm of $\mathbf{w}(t)$, normalized so it would equal to 1 at the last iteration, to facilitate comparison; **(C)** The training loss; and **(D&E)** the angle and margin gap of $\mathbf{w}(t)$ from $\hat{\mathbf{w}}$. As can be seen in panels **(C-E)**, normalized GD converges to the max-margin separator significantly faster, as expected from our results. More details are given in appendix I.1.

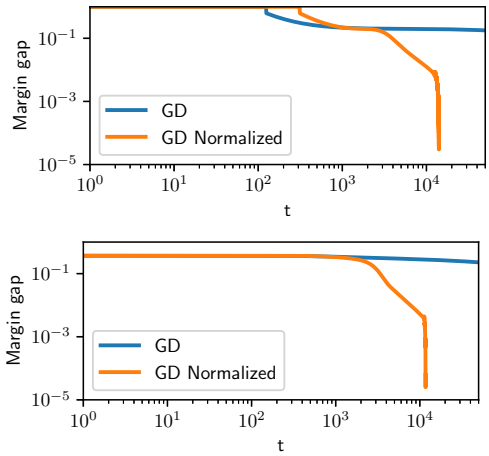

Figure 2: Margin convergence plots for 2 (top) and 3 (bottom) layered linear networks on synthetic clustered data, trained with GD and normalized GD — the latter provides significantly faster convergence.

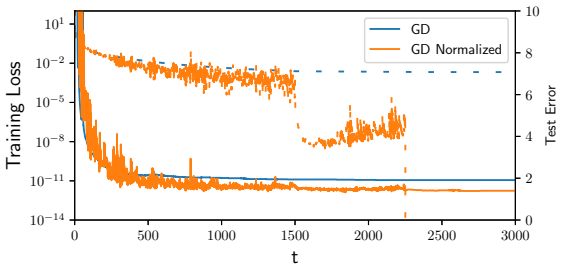

Figure 3: MNIST digit classification with a 2-layer feedforward neural network. Training loss (dashed lines) stagnates with GD once gradients become small, while normalized GD keeps making progress. Normalized GD also achieves lower test error (solid lines).

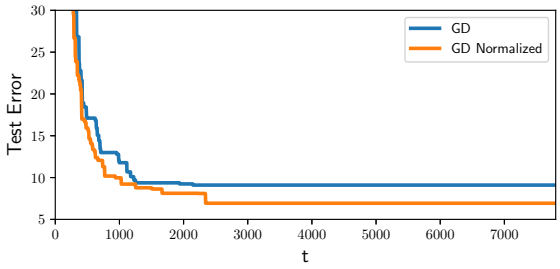

Figure 4: Test performance of a Wide ResNet 28-4 on CIFAR-10, with $\eta = 2.0$, where normalized GD outperforms GD by absolute 2.17%. We plot 'best yet' test error: the lowest error seen up to iteration t. Unlike curves reported in Zagoruyko and Komodakis (2016), progress stops early in training: there is no change in the 'best yet' test error after $t = 2350$, even with the decays in learning rate. This suggests that regularization and/or momentum might be required to achieve better results.

yielded better results for both GD and normalized GD. We use no regularization nor data augmentation, since our goal is to observe the contrast between GD and normalized GD as the training loss decreases and gradients become small. Figure 3 shows the training loss and test error at each iteration t: while the training loss stagnates early for GD, normalized GD keeps decreasing it. Normalized GD also reaches lower test error: 1.4% compared to 1.91%.

4.3 Image Classification on CIFAR-10

The CIFAR-10 dataset (Krizhevsky, 2009) consists of 60,000 colored 32×32 images belonging to one of 10 possible classes, and is split into 50,000 training and 10,000 test points. The goal of this experiment is to evaluate whether normalized GD can provide advantages for train-

ing complex models on more realistic tasks, when using the standard cross-entropy loss.

For that, we train a Wide ResNet 28-4 (Zagoruyko and Komodakis, 2016), a 28-layer convolutional neural network with residual connections and a total of 5.8M parameters. This architecture is capable of reaching less than 4% test error on CIFAR-10 given more features per convolutional layer, making Wide ResNets a strong model baseline to compare the benefits of normalized GD against GD. Following Zagoruyko and Komodakis (2016), we pre-process the dataset by performing channel-wise normalization on each image using statistics computed from the training set. Horizontal flips and random crops are used during training for data-augmentation. We also follow the same learning rate schedule, decaying it by a factor of 5 at 30%, 60% and 80% of the total iterations.

To select a learning rate for each method, we train the network for 3,000 iterations with $\eta \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$. Both methods performed better on a validation set of 5000 images with $\eta = 2.0$. Figure 4 shows the test performance when training the model for 7,800 iterations with $\eta = 2.0$, where normalized GD achieves 6.93% test error, while GD yields 9.90%.

Note that, while normalized GD outperformed GD in this full-batch setting, its performance is still subpar when compared to the standard optimization for Wide ResNets, which includes SGD with Nesterov momentum and weight decay. To confirm whether momentum and weight decay can have strong positive impacts in a model's performance, we also trained a Wide ResNet 28-4 using SGD, with and without momentum/weight decay. We observed that removing momentum and weight decay resulted in a test error increase from 4.45% to 7.75% (larger error than normalized GD). This suggests an importance in reconciling weight decay, momentum and gradient normalization.

5 DISCUSSION

In this work, we have examined the behavior of gradient descent on separable data, in binary linear classification tasks. First, in Theorem 2 we proved the linear classifier resulting from a multilayer linear neural networks converges in direction to the L_2 max-margin on almost all linearly separable data — for a wide family of monotone, convex loss functions with super-exponential tails and some technical conditions (Assumption 1). In contrast, polynomially tailed loss function do not lead to convergence to the max-margin. Intuitvely, the reason behind this is that for super-polynomial loss functions the datapoints with the largest margin (i.e., the support vectors) become dominant in the gradient, while for polynomial or heavier tails the contribution of non-support vectors is never negligible.

Next, we examine the convergence rate for a linear classifier with loss within this wide family of loss functions. We prove in Theorem 3 that the exponential tail has the optimal rate. This offers a possible explanation to the empirical preference of the exponentially-tailed loss functions over other losses (e.g. the probit loss): that the exponential loss leads to a faster convergence to the asymptotic (implicitly biased) solution, as we showed here. This result is somewhat surprising, and we do not have an intuitive explanation why this should be true.

In Theorem 4, we extend these results to multilayer linear neural networks, and show similar convergence rates, with only a negligible decrease in the rate with the depth — even when the number of layers is infinite. Note that in this Theorem we already assume convergence of the loss to zero. However, if we do converge, it is somewhat surprising that this rate does not depend much on the depth, as one might expect to have convergence rate issues due to exploding or vanishing gradients.

In Theorem 5 we showed that the convergence of GD for an exponential loss function could be significantly accelerated by simply increasing the learning rate. In fact, GD can also approximate the regularization path in the following sense. Let $R = \|\mathbf{w}_t\|$, and $\mathbf{w}_R = \arg\min_{\|\mathbf{w}\| \leq R} \mathcal{L}(\mathbf{w})$. Then

$$\mathcal{L}(\mathbf{w}(t)) - \mathcal{L}(\mathbf{w}_R) \leq \mathcal{L}(\mathbf{w}(0)) \exp(-c\gamma^2 t)\,. \qquad (12)$$

As a simple implication of this, the normalized GD path starting at $\mathbf{w}_0 = 0$ has $\mathcal{L}(\mathbf{w}(0)) = n$, so after $t \geq \log(n/\epsilon)/\gamma^2$ steps the loss achieved by $\mathbf{w}_t$ is ϵ close to the best predictor of the same norm. This shows that GD is closely approximating the regularization path.

Finally, we show numerically that normalized GD can significantly improve the convergence speed of GD on synthetic datasets for linear predictors (Figure 1), linear multilayer networks (Figure 2), and even non-linear ReLU multilayer networks (Appendix I.2). Additionally, we show normalized GD can improve the results of GD on standard datasets such as MNIST (by 0.5%) and CIFAR-10 (by 3%). However, a gap remains from achieving state of the art results. Our experiments indicate the origin of this gap is the use of weight decay and momentum (which are outside the scope of this paper). This suggests that reconciling regularization, momentum and gradient normalization might be of particular interest for future work, possibly reducing the gap between mini-batch and full-batch training.

Recent work explore extensions of the implicit bias result for linear models to non-strictly-separable datasets (Ji and Telgarsky, 2018) and to stochastic gradient descent (Ji and Telgarsky, 2018; Nacson et al., 2018; Xu et al., 2018). It remains to be seen if the results of this work could be also extended to such settings. Additionally, combining our results with the results of a parallel work, Ji and Telgarsky (2019), might enable us to weaken some of the assumptions in this paper. We discuss Ji and Telgarsky (2019) work in appendix A.

Acknowledgements

The authors are grateful to C. Zeno, and N. Merlis for helpful comments on the manuscript. This research was supported by the Israel Science foundation (grant No. 31/1031), and by the Taub foundation. A Titan Xp used for this research was donated by the NVIDIA Corporation. PS, SG and NS were partially supported by NSF awards IIS-1302662 and IIS-1764032.

References

Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):23, 2012.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018a.

Suriya Gunasekar, Jason D. Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint*, 2018b.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *NIPS*, 2017.

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.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Kfir Y. Levy. The Power of Normalization: Faster Evasion of Saddle Points. *arXiv*, nov 2016.

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*, 2018.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *International Conference on Learning Representations*, 2015.

Saharon Rosset, Ji Zhu, and Trevor J Hastie. Margin maximizing loss functions. In *Advances in neural information processing systems*, pages 1237–1244, 2004.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. MIT press, 2012.

Daniel Soudry, Elad Hoffer, , Mor Shpigel Nacson, and Nathan Srebro. The implicit bias of gradient descent on separable data. *ICLR*, 2018a.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data (journal version). *arXiv preprint: 1710.10345v3*, 2018b.

Matus Telgarsky. Margins, shrinkage and boosting. In *Proceedings of the 30th International Conference on International Conference on Machine Learning-Volume 28*, pages II–307. JMLR. org, 2013.

Willie Wong. Asymptotic solution for a first order ode. MathOverflow, 2018. URL `https://mathoverflow.net/q/309520`. URL:https://mathoverflow.net/q/309520 (version: 2018-08-31).

Tengyu Xu, Yi Zhou, Kaiyi Ji, and Yingbin Liang. Convergence of sgd in learning relu models with separable data. *arXiv preprint arXiv:1806.04339*, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

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