The Implicit Regularization for Adaptive Optimization Algorithms on Homogeneous Neural Networks

Bohan Wang [1] **Qi Meng** [1] **Wei Chen** [1] **Tie-Yan Liu** [1]

Abstract

Despite their overwhelming capacity to overfit, deep neural networks trained by specific optimization algorithms tend to generalize well to unseen data. Recently, researchers explained it by investigating the implicit regularization effect of optimization algorithms. A remarkable progress is the work (Lyu & Li, 2019), which proves gradient descent (GD) maximizes the margin of homogeneous deep neural networks. Except GD, adaptive algorithms such as AdaGrad, RMSProp and Adam are popular owing to their rapid training process. However, theoretical guarantee for the generalization of adaptive optimization algorithms is still lacking. In this paper, we study the implicit regularization of adaptive optimization algorithms when they are optimizing the logistic loss on homogeneous deep neural networks. We prove that adaptive algorithms that adopt exponential moving average strategy in conditioner (such as Adam and RMSProp) can maximize the margin of the neural network, while AdaGrad that directly sums historical squared gradients in conditioner can not. It indicates superiority on generalization of exponential moving average strategy in the design of the conditioner. Technically, we provide a unified framework to analyze convergent direction of adaptive optimization algorithms by constructing novel *adaptive gradient flow* and *surrogate margin*. Our experiments can well support the theoretical findings on convergent direction of adaptive optimization algorithms.

1. Introduction

Deep learning techniques have been very successful in several domains, like computer vision (Voulodimos et al., 2018), speech recognition (Deng et al., 2013) and natural language processing (Young et al., 2018). In practice, deep neural networks (DNN) learned by optimization algorithms such as gradient descent (GD) and its variants can generalize well to unseen data (Witten & Frank, 2005). However, deep neural networks are non-convex. The non-convex deep neural networks have been found to have large amount of global minima (Choromanska et al., 2015), while only few of them can guarantee satisfactory generalization property (Brutzkus et al., 2018). Explaining why the highly non-convex model trained by a specific algorithm can generalize has become an important open question in deep learning.

Regarding the above question, one plausible explanation is that optimization algorithms implicitly regularize the training process (Neyshabur et al., 2015). That is, the optimization algorithm tends to drive parameters to certain kinds of global minima which generalize well, although no explicit regularization is enforced. Recently, exciting results have been shown for vanilla gradient descent. A remarkable progress is the work (Lyu & Li, 2019), which proves that GD maximizes the margin of homogeneous (non-linear) deep neural networks.

On the other hand, adaptive algorithms such as AdaGrad (Duchi et al., 2011), RMSProp (Hinton et al., 2012), and Adam (Kingma & Ba, 2015) have been in spotlight these years. These algorithms are proposed to improve the convergence rate of GD (or SGD) by using second-order moments of historical gradients as conditioner and have been widely applied in deep learning (Ruder, 2016). Despite the rapid convergence of adaptive methods, numerous works have provided empirical evidence that adaptive methods may suffer from poor generalization performance (Wilson et al., 2017; Luo et al., 2018). Several works try to improve the performance of adaptive optimization algorithms such as AdamW (Loshchilov & Hutter, 2018), AdaBound (Luo et al., 2018), AdaBelief (Zhuang et al., 2020). However, there is little theoretical analysis for generalization of adaptive algorithms. These observations and the research for GD motivate us to study the implicit regularization for adaptive algorithms.

The key factor for the success of adaptive optimization algorithms is to design better conditioners of the gradient. Adagrad adopts the simple average of the squared values of the historical gradients in its conditioner, while RMSProp

[1] Microsoft Research Asia, Beijing, China. Correspondence to: Wei Chen <wche@microsoft.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, PMLR 139, 2021.

and Adam improve the simple average to exponential moving average strategy. In this paper, we aim to study the influence of different types of conditioners on convergent direction of parameters trained by adaptive optimization algorithms. Specifically, we work on the homogeneous neural networks (including fully connected or convolutional neural network with ReLU or leaky ReLU activations) with separable data under logistic loss (for binary classification) and cross-entropy (for multi-class classification). For logistic loss, we focus on characterizing the convergent direction of parameters (i.e., $\lim_{t\to\infty} \frac{w_t}{\|w_t\|_2}$) with respect to the training iteration t, which is a key target along this line of researches (Soudry et al., 2018; Gunasekar et al., 2018b; Lyu & Li, 2019).

Our main result is summarized in Theorem 1, which states that RMSProp and Adam (w/m) (a variant of Adam without momentum acceleration)[1] maximize margin of the neural network (equivalent to the optimum of optimization problem in Eq.(2)) and AdaGrad does not converge to max-margin solution due to the anisotropic $\boldsymbol{h}_\infty$.

Theorem 1. *(Informal) We use $\Phi(\boldsymbol{w}, \boldsymbol{x})$ to denote the homogeneous neural network model with parameter $\boldsymbol{w}$ and input $\boldsymbol{x}$. (1) For AdaGrad, any limit point of $w_t/\|w_t\|_2$ is a KKT point of the optimization problem*

$$\min \|\boldsymbol{h}_\infty^{-1/2} \odot \boldsymbol{w}\|^2 \quad \textit{subject to } y_i\Phi(\boldsymbol{w}, \boldsymbol{x}_i) \geq 1, \forall i, \quad (1)$$

where $\boldsymbol{h}_\infty = \lim_{t\to\infty} \boldsymbol{h}(t)$ is the limit of the conditioner in AdaGrad. (2) For Adam (w/m) and RMSProp, any limit point of $\boldsymbol{w}(t)/\|\boldsymbol{w}(t)\|_2$ is a KKT point of the optimization problem

$$\min \|\boldsymbol{w}\|^2 \quad \textit{subject to } y_i\Phi(\boldsymbol{w}, \boldsymbol{x}_i) \geq 1, \forall i. \quad (2)$$

Theorem 1 indicates the importance of proper design on the conditioner, i.e., adaptive algorithms like Adam (w/m) and RMSProp that adopt exponential weighted average design on conditioner regularize the training to max-margin solution, which has low complexity. Therefore, we can expect good generalization performance for Adam (w/m) and RMSProp. Furthermore, we illustrate that the convergence direction of AdaGrad is sensitive to initialization, which hurts its generalization.

We establish Theorem 1 for both continuous flows of adaptive optimization algorithms and their discrete update rules. The **technical contributions** to prove Theorem 1 are summarized as follows. (1) We propose *adaptive gradient flow*, which is a unified framework to deal with adaptive gradients. With the adaptive gradient flow, the analysis of convergent direction is transformed from original parameter space to a normalized parameter space. (2) In the normalized parameter space, we construct *surrogate margin* for the adaptive algorithms, and with the surrogate margin, we show that the increasing rate of the parameter norm can be bounded by the decreasing rate of logarithmic loss and the loss converges to zero. (3) We prove that any limit direction of the normalized parameter flow is a KKT point of the margin maximization problem in normalized parameter space. Moreover, we prove the convergent direction is unique if the neural network is definable (Kurdyka, 1998). The adaptive gradient flow and surrogate margin are designed for adaptive optimization algorithms, which makes the proof techniques different from that for vanilla GD in (Soudry et al., 2018; Lyu & Li, 2019). (4) We further prove the convergent direction for discrete update rules by characterizing the influence of the learning rate.

Finally, we conduct experiments to observe the margin of homogeneous neural network during training of several adaptive optimization algorithms. For all experiments, the margins are increasing during training and the final margins of RMSProp and Adam (w/m) are larger than that of AdaGrad. We also observe the convergent direction of adaptive optimization algorithms under different realizations of initialization and results show that the convergent direction of AdaGrad is sensitive to initialization. These observations can well support our theoretical findings.

2. Related Work

Implicit Regularization of First-order Optimization Methods. Soudry et al. (2018) proved that gradient descent on linear logistic regression with separable data converges in the direction of the max L^2 margin solution of the corresponding hard-margin Support Vector Machine, and motivate a line of works on the implicit regularization of GD on linear model (Nacson et al., 2019b; Ji & Telgarsky, 2019; Li et al., 2019; Xu et al., 2018).

Afterwards, researchers study the implicit regularization of GD on deep neural networks. Ji & Telgarsky (2018); Gunasekar et al. (2018b) studied the deep linear network and Soudry et al. (2018) studied the two-layer neural network with ReLU activation. Nacson et al. (2019a) proved the asymptotic direction is along a KKT point of the L^2 max-margin problem for homogeneous deep neural networks. Lyu & Li (2019) independently proved similar result for homogeneous neural networks with simplified assumptions. Based on (Lyu & Li, 2019), Ji & Telgarsky (2020) further prove that parameters have only one asymptotic direction.

There are also works considering implicit regularization of other first-order optimization algorithms. Nacson et al. (2019c) worked on Stochastic Gradient Descent for linear

[1] How momentum influence the convergence of an optimization algorithm on non-convex deep neural network is still an open problem. Here, we only study a variant of Adam which sets the momentum parameter as 0.

logistic regression. Gunasekar et al. (2018a) studied mirror descent and steepest descent on linear model. Arora et al. (2019) proved gradient descent on Neural Tangent Kernel will converge to a global minimum near the initial point.

However, there is little result on the implicit regularization of adaptive optimization methods.

Theoretical Evidence of Generalization of Adaptive Algorithms. Adaptive algorithms have been in spotlight these years and many works empirically observe the generalization behavior of adaptive algorithms (Keskar & Socher, 2017; Reddi et al., 2018; Chen et al., 2018; Luo et al., 2018). In comparison, there are few theoretical justifications. Wilson et al. (2017) constructed a specific linear regression task where adaptive optimization algorithms converge to a solution that incorrectly classifies new data with probability arbitrarily close to half. Zhou et al. (2020) modeled the distribution of stochastic noise in Adam, and showed that SGD tends to converge to flatter local minima. Another viewpoint is to study **the convergent direction of adaptive optimization algorithms**. To the best of our knowledge, the only work is (Qian & Qian, 2019), which proves the convergent direction of AdaGrad on linear logistic regression. In this paper, we study the convergent direction of adaptive optimization algorithms on deep neural networks which requires different techniques due to the non-convexity of deep networks.

Meanwhile, the **correlation between margin and generalization error** has also been extended to deep networks. Bartlett et al. (2017) first bound the generalization error of deep neural networks using (spectrally) normalized margin by covering number. In parallel, Neyshabur et al. (2018) adopt normalized margin into the PAC-Bayesian framework and derive generalization bound with different dependency on layer width from (Bartlett et al., 2017). Empirically, Jiang et al. (2019) present a large scale study of different generalization bounds in deep networks, and find there is a significant correlation between generalization error and normalized margin when optimizer is changed. These work support our study on generalization in deep learning through the margin theory.

3. Preliminaries

In this paper, we study the logistic regression problem with homogeneous neural networks. Let training set $\boldsymbol{S}$ defined as $\boldsymbol{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ $(i = 1, 2, \cdots, N)$ are inputs, $y_i \in \mathbb{R}$ $(i = 1, 2, \cdots, N)$ are labels, and N is the size of $\boldsymbol{S}$. The empirical loss $\mathcal{L}$ with training set $\boldsymbol{S}$, neural network classifier Φ, individual loss $\ell(x) = e^{-f(x)}$ and parameters $\boldsymbol{w} \in \mathbb{R}^p$ can be written as follows:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{S}) = \sum_{i=1}^{N} \ell\left(y_i \Phi(\boldsymbol{w}, \boldsymbol{x}_i)\right).$$

In an optimization process, the training set $\boldsymbol{S}$ is fixed. Therefore, without loss of generality, we abbreviate $\mathcal{L}(\boldsymbol{w}, \boldsymbol{S}) = \mathcal{L}(\boldsymbol{w})$, and $y_i \Phi(\boldsymbol{w}, \boldsymbol{x}_i) = q_i(\boldsymbol{w})$. In this paper, we consider the exponential loss, i.e., $f(q_i(\boldsymbol{w})) = q_i(\boldsymbol{w})$, and the logistic loss, i.e., $f(q_i(\boldsymbol{w})) = -\log\log(1 + e^{-q_i(\boldsymbol{w})})$. Both of f are monotonously increasing and have an inverse.

We will use Clarke's Subdifferential $\bar{\partial}$ (Clarke, 1975) in this paper as a natural extension of gradient ∇ for locally Lipschitz functions. For any locally Lipschitz function $f : \mathbb{R}^p \to \mathbb{R}$, its Clarke's Subdifferential $\bar{\partial} f$ at point $\boldsymbol{w}_0$ is defined as

$$\operatorname{conv}\{\lim_{k\to\infty} \nabla f(\boldsymbol{w}_k) : \boldsymbol{w}_k \to \boldsymbol{w}_0, \nabla f(\boldsymbol{w}_k) \text{ exists } \}.$$

Following (Davis et al., 2020), we also define f admits a chain rule if for any arc [2] $\boldsymbol{z} : \mathbb{R}^+ \to \mathbb{R}^p$, $\forall \boldsymbol{h} \in \bar{\partial} f(\boldsymbol{z}(t))$, $\frac{\mathrm{d}f(z(t))}{\mathrm{d}t} = \langle \boldsymbol{h}, \frac{\mathrm{d}z}{\mathrm{d}t} \rangle$.

3.1. Continuous Flow for Adaptive Algorithms

Adaptive optimization algorithms including AdaGrad, RMSProp, Adam are widely used to optimize the loss function in deep learning. The update rules for these adaptive optimization algorithms can be written as [3]

$$\boldsymbol{w}(k+1) - \boldsymbol{w}(k) = -\eta \boldsymbol{h}(k) \odot \bar{\partial}\mathcal{L}(\boldsymbol{w}(k)), \tag{3}$$

where $k = 1, 2, \cdots$ denotes the iteration index, η denotes a constant learning rate, $\boldsymbol{h}(k)$ is called the conditioner which adaptively assigns different learning rates for different coordinates. For AdaGrad, $\boldsymbol{h}(k)^{-1} = \sqrt{\varepsilon \mathbf{1}_p + \sum_{\tau=0}^{k} \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2}$ where ϵ is a positive constant, and $\mathbf{1}_p$ is a length-p vector with all components to be 1. Here, $\bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2 = \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau)) \odot \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))$ and $\odot$ denotes the element-wise product of a vector. Different from AdaGrad, RMSProp adopts exponential weighted average strategy in $\boldsymbol{h}(k)$, i.e., $\boldsymbol{h}(k)^{-1} = \sqrt{\varepsilon \mathbf{1}_p + \sum_{\tau=0}^{k} (1-b) b^{k-\tau} \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2}$. Adam further introduces a bias-correction coefficient $\frac{1}{1-b^k}$ and $\boldsymbol{h}(k)^{-1} = \sqrt{\varepsilon \mathbf{1}_p + \frac{\sum_{\tau=0}^{k} (1-b) b^{k-\tau} \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2}{1-b^k}}$. In this paper, we use $\boldsymbol{h}^A(k)$, $\boldsymbol{h}^R(k)$ and $\boldsymbol{h}^M(k)$ to distinguish the term $\boldsymbol{h}(k)$ in AdaGrad, RMSProp and Adam respectively.

Taking $\eta \to 0$, the continuous time limits (i.e., continuous flow) of the three optimization algorithms are

$$\frac{d\boldsymbol{w}(t)}{\mathrm{d}t} = -\boldsymbol{h}(t) \odot \bar{\partial}\mathcal{L}(\boldsymbol{w}(t)), \tag{4}$$

$\boldsymbol{h}^A(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_p + \int_0^t \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2 \mathrm{d}\tau}$, $\boldsymbol{h}^R(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_p + \int_0^t (1-b) e^{-(1-b)(t-\tau)} \bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2 \mathrm{d}\tau}$ and

[2] A arc $\boldsymbol{z} : \mathbb{R}^+ \to \mathbb{R}^p$ satisfies for any compact set $I \subset \mathbb{R}^+$, $\boldsymbol{z}$ is absolute continuous on I.

[3] In this paper, we only consider no-momentum versions of the algorithms, i.e., the algorithms without momentum acceleration.

$\boldsymbol{h}^M(t)^{-1} = \sqrt{\varepsilon \mathbf{1}_p + \frac{\int_0^t (1-b)e^{-(1-b)(t-\tau)}\bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2 \mathrm{d}\tau}{1-b^t}}$.

Our study will start with the continuous version of the two algorithms. Specifically, for the continuous case, we focus on the following scenario.

Assumption 1. *The empirical loss is defined as* $\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^N e^{-f(q_i(\boldsymbol{w}))}$. *The following propositions hold:*

I (Regularity). For any i, $\Phi(\boldsymbol{w}, \boldsymbol{x}_i)$ *is locally Lipschitz and admits a chain rule with respect to* $\boldsymbol{w}$;

II (Homogeneity). There exists $L > 0$ *such that* $\forall \alpha > 0$ *and* i, $\Phi(\alpha \boldsymbol{w}, \boldsymbol{x}_i) = \alpha^L \Phi(\boldsymbol{w}, \boldsymbol{x}_i)$;

III (Separability). There exists a time t_0 *such that* $f^{-1}(\log \frac{1}{\mathcal{L}(t_0)}) > 0$.

I, II in Assumption 1 holds for a board class of networks allowing for ReLU, max pooling, and convolutional layers; Assumption 1.III holds generally for over-parameterized neural networks, which can achieve complete correct classification in training set.

3.2. KKT point

We give a brief introduction to KKT conditions and KKT points. For a constrained optimization problem defined as

$$\min f(\boldsymbol{w}) \text{ subject to: } g_i(\boldsymbol{w}) \leq 0,\ \forall i \in [N],$$

KKT conditions are necessary conditions for a point $\boldsymbol{w}_0$ to be optimal in above problem, which require that there exists non-negative reals λ_i, such that

$$\bar{\partial} f(\boldsymbol{w}_0) + \sum_{i=1}^N \lambda_i \bar{\partial} g_i(\boldsymbol{w}_0) = 0; \quad \sum_{i=1}^N \lambda_i g_i(\boldsymbol{w}_0) = 0. \quad (5)$$

A weaker notion of KKT condition is (ε, δ) KKT condition, which requires left sides of eq. (5) to be respectively smaller than ε and δ. We will formally define (ε, δ) KKT points and give some of their properties in Appendix A.2.

Notations. In this paper, we use $\boldsymbol{o}$, $\mathcal{O}$, Θ, and Ω to hide the absolute multiplicative factors. Concretely, $f(t) = \boldsymbol{o}(g(t))$ if $\overline{\lim}_{t\to\infty} \frac{f(t)}{g(t)} = 0$; $f(t) = \mathcal{O}(g(t))$ if $\overline{\lim}_{t\to\infty} \frac{f(t)}{g(t)} < \infty$; $f(t) = \Omega(g(t))$ if $\underline{\lim}_{t\to\infty} \frac{f(t)}{g(t)} > 0$; $f(t) = \Theta(g(t))$ if $f(t) = \Omega(g(t))$ and $f(t) = \mathcal{O}(g(t))$.

4. Main Results

In this section, we introduce the main results on convergent direction of adaptive optimization algorithms. In Section 4.1, we propose a unified adaptive gradient flow and prove that it converges to KKT point of max-margin problem. In Section 4.2, we apply results for adaptive gradient flow to AdaGrad, RMSProp and Adam (w/m) to get the convergent directions of their continuous flow. In Section 4.3, we prove the convergent directions of the discrete update rules of adaptive optimization algorithms.

4.1. Adaptive Gradient Flow: Definition and Results

Adaptive optimizers such as AdaGrad, RMSProp and Adam can be viewed as adding component-wise conditioner to gradient updates and the limit of the component-wise conditioner may be anisotropic for different components. We first define adaptive gradient flow whose limit of component-wise conditioner is isotropic.

Definition 1. *A function* $\boldsymbol{w}(t)$ *is called to obey an adaptive gradient flow* $\mathcal{F}$ *with loss* $\mathcal{L}$ *and component learning rate* $\boldsymbol{\beta}(t)$, *if it can be written as the following form*

$$\frac{d\boldsymbol{v}(t)}{\mathrm{d}t} = -\boldsymbol{\beta}(t) \odot \bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v}(t)),$$

where $\boldsymbol{\beta}(t)$ *satisfies that* $\lim_{t\to\infty} \boldsymbol{\beta}(t) = \mathbf{1}_p$, *and* $\frac{\mathrm{d}\log \boldsymbol{\beta}(t)}{\mathrm{d}t}$ *is Lebesgue Integrable.*

We make some explanations for Definition 1: Conditions $\lim_{t\to\infty} \boldsymbol{\beta}(t) = \mathbf{1}_p$ and $\frac{\mathrm{d}\log \boldsymbol{\beta}(t)}{\mathrm{d}t}$ being Lebesgue Integrable ensures $\boldsymbol{\beta}(t)$ converges to $\mathbf{1}_p$ without large fluctuation. These constraints are common, in the sense that AdaGrad, RMSProp and Adam (w/m) can be transferred into such flows by simple reparameterization (see Section 4.2); but are also vital, which guarantee adaptive gradient flows converge to KKT point of max-margin problem as follows:

Theorem 2. *Let* $\boldsymbol{v}$ *obey an adaptive gradient flow* $\mathcal{F}$ *which satisfies Assumption 1. Let* $\bar{\boldsymbol{v}}$ *be any limit point of* $\{\hat{\boldsymbol{v}}(t)\}_{t=0}^{\infty}$ *(where* $\hat{\boldsymbol{v}}(t) = \frac{\boldsymbol{v}(t)}{\|\boldsymbol{v}(t)\|}$ *). Then* $\bar{\boldsymbol{v}}$ *is along the direction of a KKT point of the following* L^2 *max-margin problem* (P)*:*

$$\min \frac{1}{2}\|\boldsymbol{v}\|^2$$
$$\text{subject to } \tilde{q}_i(\boldsymbol{v}) \geq 1, \forall i \in [N].$$

(P) is equivalent to the L^2 max-margin problem: suppose $\boldsymbol{v}_0$ is an optimal point of (P). Then there exists an $i \in [N]$, such that, $\tilde{q}_i(\boldsymbol{v}_0) = 1$ (otherwise, we can let $\boldsymbol{v_0}' = \boldsymbol{v}_0/\tilde{q}_{\min}(\boldsymbol{v}_0)^{\frac{1}{L}}$. Then $\boldsymbol{v_0}'$ is also a fixed point of (P) and have a smaller L^2 norm than $\boldsymbol{v_0}'$, which leads to contradictory). Therefore, $\tilde{q}_{\min}(\boldsymbol{v}_0) = 1$ ($\tilde{q}_{\min}(\boldsymbol{v}) \triangleq \min_i\{\tilde{q}_i(\boldsymbol{v})\}$), and maximizing the normalized margin $\frac{\tilde{q}_{\min}(\boldsymbol{v})}{\|\boldsymbol{v}\|^L}$ is equivalent to minimize $\|\boldsymbol{v}\|^2$.

Theorem 2 shows that the adaptive gradient flow actually drives the parameters to solutions of L^2 max-margin problem. We will give the proof skeleton of Theorem 2 in Section 5.

Remark 1. *Our result can be extended to the multi-class classification with logistic loss and same assumption as Assumption 1 except that $\Phi(\boldsymbol{w}, \boldsymbol{x}_i)$ is a C-dimension vector in multi-class case with C number of classes. The corresponding L^2 max-margin classification problem is then*

$$\min \frac{1}{2}\|\boldsymbol{v}\|^2$$
$$\text{subject to } (\Phi(\boldsymbol{w}, \boldsymbol{x}_i))_{y_i} - (\Phi(\boldsymbol{w}, \boldsymbol{x}_i))_j \geq 1,$$
$$\forall i \in [N], j \in [C]/\{y_i\}.$$

We defer the proof to Appendix E.

While Theorem 2 does NOT guarantee direction of parameters converges as $t \to \infty$, we present a theorem in the end of this section which provides such a guarantee when neural network Φ is definable with respect to parameters $\boldsymbol{w}$.

Theorem 3. *Let all assumptions in Theorem 2 hold. Assume further $\Phi(\boldsymbol{w}, \boldsymbol{x}_i)$ is definable with respect to parameter $\boldsymbol{w}$ for any $i \in [N]$. Then direction of parameters $\{\hat{\boldsymbol{v}}(t)\}_{t=0}^{\infty}$ converges.*

We defer the formal definition of definable to Appendix C, but point out here that definability allows for linear, ReLU, polynomial activations, max pooling and convolutional layers, and skip connections. Furthermore, for locally Lipschitz definable function, chain rule holds almost everywhere (Lemma 11).

The proof can be derived by bounding the curve length of $\hat{\boldsymbol{v}}(t)$ using $\tilde{\gamma}(t)$ and Kurdyka-Lojasiewicz inequalities developed in (Ji & Telgarsky, 2020), and we defer the details to Appendix C.

4.2. Results for Adaptive Algorithms: Continuous Case

In this section, we will prove gradient flow of AdaGrad, RMSProp, and Adam (w/m) can be transferred into adaptive gradient flow. We start from proving convergence of conditioner in AdaGrad and further shows AdaGrad can be reparameterized as an adaptive gradient flow.

Theorem 4. *For AdaGrad flow defined as eq. (4) with $\boldsymbol{h}(t) = \boldsymbol{h}^A(t)$, we have that*

- *$\boldsymbol{h}^A(t)$ converges as $t \to \infty$. Furthermore, $\boldsymbol{h}_\infty = \lim_{t\to\infty} \boldsymbol{h}^A(t)$ has no zero component.*
- *$\frac{\mathrm{d}\boldsymbol{v}^A(t)}{\mathrm{d}t} = -\boldsymbol{\beta}^A(t) \odot \bar{\partial}\tilde{\mathcal{L}}^A(\boldsymbol{v}^A(t))$ satisfies definition of adaptive gradient flow, where*

$$\boldsymbol{v}^A(t) = \boldsymbol{h}_\infty^{-1/2} \odot \boldsymbol{w}(t), \boldsymbol{\beta}^A(t) = \boldsymbol{h}_\infty^{-1} \odot \boldsymbol{h}^A(t),$$
$$\tilde{\mathcal{L}}^A(\boldsymbol{v}^A) = \mathcal{L}(\boldsymbol{h}_\infty^{\frac{1}{2}} \odot \boldsymbol{v}^A).$$

We provide some intuitions for proof of Theorem 4. The former part of the first property is because $\boldsymbol{h}^A(t)$ is non-increasing with respect to t. However, the latter part yields that integration of square of the gradient converges to a positive real, which is non-trivial; the second property is obtained by component-wisely scaling $\boldsymbol{w}$ and direct verification; the last property can be obtained by Newton-Leibniz formula for absolutely continuous function since $\frac{\mathrm{d}\boldsymbol{\beta}^A(t)}{\mathrm{d}t}$ is non-negative. We defer the detailed proof to Appendix B.1.

Similar properties also hold for RMSProp and Adam (w/m) as the following Theorem .

Theorem 5. *For RMSProp and Adam flow defined as eq. (4) respectively with $\boldsymbol{h}(t) = \boldsymbol{h}^R(t)$ and $\boldsymbol{h}(t) = \boldsymbol{h}^M(t)$, we have that, for $I \in \{R, M\}$,*

- *$\boldsymbol{h}^I(t)$ converges as $t \to \infty$. Furthermore, $\lim_{t\to\infty} \boldsymbol{h}^I(t) = \varepsilon^{-\frac{1}{2}} \mathbf{1}_p^T$.*
- *$\frac{\mathrm{d}\boldsymbol{v}^I(t)}{\mathrm{d}t} = -\boldsymbol{\beta}^I(t) \odot \bar{\partial}\tilde{\mathcal{L}}^I(\boldsymbol{v}^A(t))$ satisfies definition of adaptive gradient flow, where*

$$\boldsymbol{v}^I(t) = \varepsilon^{\frac{1}{4}} \boldsymbol{w}(t), \boldsymbol{\beta}^I(t) = \varepsilon^{\frac{1}{2}} \boldsymbol{h}(t), \tilde{\mathcal{L}}^I(\boldsymbol{v}^I) = \mathcal{L}(\varepsilon^{-\frac{1}{4}} \boldsymbol{v}^I).$$

Both conditioners $\boldsymbol{h}^R$ and $\boldsymbol{h}^M$ have an exponential decay term $e^{-(t-\tau)(1-b)}$, which drives $\int_{\tau=0}^{t}(1-b)e^{-(1-b)(t-\tau)}\bar{\partial}\mathcal{L}(\boldsymbol{w}(\tau))^2 d\tau$ to zero, and conditioners to isotropy. The detailed proof requires a more careful analysis in measure than the AdaGrad flow. We defer them to Section B.1.

By Theorems 4 and 5, gradient flow of AdaGrad, RMSProp and Adam (w/m) can both be transferred into adaptive gradient flows: $\boldsymbol{v}^I$ obeys an adaptive gradient flow with loss $\tilde{\mathcal{L}}^I$ and conditioner $\boldsymbol{\beta}^I$ ($I \in A, R, M$). Furthermore, Assumption 1 also holds for $\tilde{\mathcal{L}}^I$: for AdaGrad, RMSProp, and Adam , we can uniformly represent $\tilde{\mathcal{L}}^A(\boldsymbol{v}^A)$, $\tilde{\mathcal{L}}^R(\boldsymbol{v}^R)$, and $\tilde{\mathcal{L}}^M(\boldsymbol{v}^M)$ as $\mathcal{L}(\tilde{\boldsymbol{h}}^{\frac{1}{2}} \odot \boldsymbol{v})$, where $\tilde{\boldsymbol{h}}$ is a component-wisely positive constant vector. By Assumption 1, $\tilde{\mathcal{L}}(\boldsymbol{v})$ can be further written as

$$\tilde{\mathcal{L}}(\boldsymbol{v}) = \mathcal{L}(\tilde{\boldsymbol{h}}^{\frac{1}{2}} \odot \boldsymbol{v}) = \sum_{i=1}^{N} e^{-f(q_i(\tilde{\boldsymbol{h}}^{\frac{1}{2}} \odot \boldsymbol{v}))}.$$

If we denote $\tilde{q}_i(\boldsymbol{v}) = q_i(\tilde{\boldsymbol{h}}^{\frac{1}{2}} \odot \boldsymbol{v})$, we have $\tilde{q}_i$ is also an L homogeneous function, and $\tilde{\mathcal{L}}(\boldsymbol{v}) = \sum_{i=1}^{N} e^{-f(\tilde{q}_i(\boldsymbol{v}))}$.

Combining Theorem 2 with Theorems 4 and 5, one can obtain convergent directions of AdaGrad flow and RMSProp flow by simple parameter substitution of (P).

Theorem 6. *Let $\boldsymbol{w}$ satisfy AdaGrad flow defined as eq. (4) with $\boldsymbol{h}(t) = \boldsymbol{h}^A(t)$. Then, any limit point of $\{\hat{\boldsymbol{w}}(t)\}_{t=0}^{\infty}$ (where $\hat{\boldsymbol{w}}(t) = \frac{\boldsymbol{w}(t)}{\|\boldsymbol{w}(t)\|}$ is normalized parameter) is along the direction of a KKT point of the following optimization problem (P^A):*

$$\min \frac{1}{2}\|\boldsymbol{h}_\infty^{-\frac{1}{2}} \odot \boldsymbol{w}\|^2$$
$$\text{Subject to: } q_i(\boldsymbol{w}) \geq 1.$$

Theorem 7. *Let $\boldsymbol{w}$ satisfy RMSProp or Adam flow defined as eq. (4) respectively with $\boldsymbol{h}(t) = \boldsymbol{h}^R(t), \boldsymbol{h}^M(t)$. Then, any limit point of $\{\hat{\boldsymbol{w}}(t)\}_{t=0}^{\infty}$ (where $\hat{\boldsymbol{w}}(t) = \frac{\boldsymbol{w}(t)}{\|\boldsymbol{w}(t)\|}$ is normalized parameter) is along the direction of a KKT point of the following optimization problem (P^R):*

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\textit{Subject to: } q_i(\boldsymbol{w}) \geq 1.$$

Intuitively, (P^R) is the L^2 max-margin problem, which means RMSProp flow biases parameters to a local minimum with good generalization property; on the other hand, the target of (P^A) has a reliance of $\boldsymbol{h}_\infty$, which is a constant vector in (P^A) but can be influenced by the optimization process and initialization, and may further lead to worse generalization. We will discuss the difference between convergent directions of AdaGrad and RMSProp in detail in Section 4.4.

4.3. Results for Adaptive Algorithms: Discrete Case

In practice, gradient descent methods are employed since calculating exact gradient flow requires huge efforts. In this section, we show same results hold in Theorems 6 and 7 for discrete update rules of adaptive algorithms with slightly different assumptions.

As for the discrete case, two additional assumptions are needed as follows (For brevity, we put the complete assumption to the appendix):

Assumption 2. *I (smooth). For any fixed x, $\Phi(\cdot; x)$ is M smooth (i.e., Φ is twice continuously differentiable with respect to x and all the eigenvalues of the Hessian are within $[-M, M]$);*

II (Learning Rate). For $k > k_0$, $\eta_t \leq C(t)$, where $C(t)$ is a non-decreasing function (defined in Appendix D). Also, η_t is lower bounded by a positive real, that is, there exists a constant $\tilde{\eta} > 0$, such that, for any $k > k_0$, $\eta_k \geq \tilde{\eta}$.

We make the following explanations for Assumption 2. Assumption 2(I) is needed technically because we need to consider second order Taylor expansion around each point along the training $\{\boldsymbol{w}(k); k = 1, \cdots, \}$. Results based on this assumption are the state-of-art in the existing literature of the implicit bias of GD (e.g. [3]). We put loosening this assumption to future works. Assumption 2(II) guarantees that the second order Taylor expansion is upper bounded and the step size is not too small. With Assumption 2, we have the following theorem:

Theorem 8. *With Assumptions 1 and Assumption 2, Theorems 6 and Theorems 7 hold respectively for discrete update of AdaGrad and discrete updates of RMSProp and Adam.*

We put the proof for Theorem 8 to Appendix D.

4.4. Discussions

We make some discussions on the results derived in Section 4.2 and 4.3. First, as shown in (Li et al., 2019), the optimization problem P^R is equivalent to L_2 margin maximization problem. Theorems 7 and 6 show that RMSProp and Adam (w/m) converge to max-margin solution, while AdaGrad may drive the parameters to a different direction. The corresponding optimization problem of AdaGrad has a reliance on $\boldsymbol{h}_\infty$, which is shown to be sensitive to the optimization path before convergence (shown in Section 6.2), and makes the convergent direction sensitive (we will discuss this in detail in Appendix A.5). Because the normalized margin is used as a complexity norm in generalization literature (i.e., larger normalized margin indicating better generalization performance) (Bartlett & Shawe-Taylor, 1999), our results indicate the superiority on generalization of exponential moving average strategy in the design of the conditioner.

Second, two key factors that guarantee generalization of RMSProp and Adam are exponential weighted average design on the conditioner and the added constant ϵ in $\boldsymbol{h}(t)$. Our results show the benefit of the two factors: it accelerates the training process at early stage of optimization by adaptively adjusting the learning rate, but it still converges to max-margin solution because the denominator of conditioner tends to constant ϵ at later stage. Most of previous works explain ϵ to ensure positivity of $\boldsymbol{h}(t)$. Our results show that ϵ is important for the convergent direction of the parameters and the generalization ability.

5. Proof Sketch of Theorem 2

In this section, we present the proof sketch of Theorem 2. The proof can be divided into three stages: (I) we define surrogate margin and prove that it is lower bounded and equivalent to normalized margin as time tends to infinity; (II) We use surrogate margin to lower bound the decreasing rate of empirical loss $\mathcal{L}$, and prove $\lim_{t\to\infty} \tilde{\mathcal{L}}(\boldsymbol{w}(t)) = 0$; (III) For every convergent direction $\bar{\boldsymbol{v}}$, a series of $(\varepsilon_i, \delta_i)$ KKT point which converges to $\bar{\boldsymbol{v}}$ with $\lim_{i\to\infty} \varepsilon_i = \lim_{i\to\infty} \delta_i = 0$ is constructed. We then show every convergent direction is a KKT point of optimization problem (P).

5.1. surrogate margin on adaptive gradient flow

For adaptive gradient flow $\bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v}(t)) = -\boldsymbol{\beta}(t) \odot \frac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t}$, we first deal with the change of $\|\boldsymbol{v}\|$. To derive change of $\|\boldsymbol{v}\|$, we study the surrogate norm $\rho(t) \triangleq \|\boldsymbol{\beta}(t)^{-\frac{1}{2}} \odot \boldsymbol{v}(t)\|$ because $\rho(t) = \Theta(\|\boldsymbol{v}(t)\|)$ based on $\lim_{t\to\infty} \boldsymbol{\beta}(t) = \mathbf{1}$.

The normalized margin $\gamma(t) = \frac{\tilde{q}_{\min}(t)}{\|\boldsymbol{v}(t)\|^L}$ connects margin $\tilde{q}_{\min}(t)$ with parameter norm $\|\boldsymbol{v}(t)\|$. The next lemma ad-

mits us to define an surrogate margin using $\tilde{\mathcal{L}}$.

Lemma 1. *If* $\lim_{t\to\infty}\tilde{\mathcal{L}}=0$, *we have* $\frac{f^{-1}(\log\frac{1}{\tilde{\mathcal{L}}(\boldsymbol{v}(t))})}{\rho(t)^L}=\Theta(\gamma(t))$.

Based on Lemma 1, we define surrogate margin $\tilde{\gamma}(t)$ as

$$\tilde{\gamma}(t)=\frac{f^{-1}(\log\frac{1}{\tilde{\mathcal{L}}(\boldsymbol{v}(t))})}{\rho(t)^L}.$$

Since $\rho(t)=\Theta(\|\boldsymbol{v}(t)\|)$, $\tilde{\gamma}$ actually bridge the norm of parameters with empirical loss. A desired property for $\tilde{\gamma}$ is to have a positive lower bound, since with this property, one can further bound parameter norm using empirical loss. The following lemma shows that $\tilde{\gamma}$ is lower bounded for adaptive gradient flow $\mathcal{F}$ with empirical loss $\tilde{\mathcal{L}}$ satisfying Assumption 1.

Lemma 2. *Let a function* $\boldsymbol{v}(t)$ *obey an adaptive gradient flow* $\mathcal{F}$ *with loss* $\tilde{\mathcal{L}}$ *and component learning rate* $\boldsymbol{\beta}(t)$, *where* $\tilde{\mathcal{L}}$ *satisfies Assumption 1. Then there exists a time* $t_1\geq t_0$, *such that, for any time* $t\geq t_1$, $\tilde{\gamma}(t)\geq e^{-\frac{1}{2}}\tilde{\gamma}(t_1)$.

Remark 2. *Our surrogate margin can be obtained by replacing* $\|\boldsymbol{v}(t)\|$ *by* $\rho(t)=\|\boldsymbol{\beta}(t)^{-\frac{1}{2}}\odot\boldsymbol{v}(t)\|$ *in the smoothed margin in [3]. This allows us to lower bound the derivative of surrogate margin and further lower bound the surrogate margin as Lemma 2, while the derivative of smoothed margin for adaptive gradient flow can not be bounded easily.*

Here we briefly give a road map of the proof. The derivative of norm $\rho(t)$ can be split into two parts: one is the increasing of parameter $\boldsymbol{v}$, and another is the change of component learning rate $\boldsymbol{\beta}^{-\frac{1}{2}}(t)$. Applying homogeneity of $\tilde{q}_i$ and Cauchy–Schwarz inequality, we bound the first term using the derivative of $f^{-1}\left(\log\left(\frac{1}{\tilde{\mathcal{L}}(t)}\right)\right)$; the second term can be lower bounded by $\sum_{i=1}^p\left(\frac{d\log\boldsymbol{\beta}_i^{-\frac{1}{2}}(t)}{\mathrm{d}t}\right)_+$, whose integration is bounded by the definition of adaptive gradient flow. The proof is completed by putting two parts together.

By the discussion above, one can conclude that derivative of $\tilde{\gamma}(t)$ can be calculated by subtracting a small enough term $\sum_{i=1}^p\left(\frac{d\log\boldsymbol{\beta}_i^{-\frac{1}{2}}(t)}{\mathrm{d}t}\right)_+$ from a non-negative term. This fact leads to the the convergence of $\tilde{\gamma}(t)$.

Lemma 3. *Suppose a function* $\boldsymbol{v}$ *obey an adaptive gradient flow* $\mathcal{F}$, *which satisfies Assumption 1. Then the surrogate margin* $\tilde{\gamma}(t)$ *converges.*

5.2. Convergence of Empirical Loss and Parameters

By Lemma 2, we have that for an adaptive gradient flow $\mathcal{F}$ with Assumption 1, the norm $\rho(t)$ can be bounded as $\rho(t)=\mathcal{O}(f^{-1}(\log(\frac{1}{\tilde{\mathcal{L}}(\boldsymbol{v}(t))}))^{\frac{1}{L}})$. On the other hand, by chain rule, the derivative of empirical loss with respect to time can be calculated as

$$\frac{\mathrm{d}\tilde{\mathcal{L}}(\boldsymbol{v}(t))}{\mathrm{d}t}=\langle\bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v}(t)),\frac{\mathrm{d}\boldsymbol{v}(t)}{\mathrm{d}t}\rangle=-\|\boldsymbol{\beta}^{\frac{1}{2}}(t)\odot\bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v}(t))\|^2$$
$$\leq-\frac{\langle\bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v}(t)),\boldsymbol{v}(t)\rangle^2}{\rho(t)^2},$$

where the last inequality is derived by the Cauchy inequality applying to $\langle\boldsymbol{\beta}^{\frac{1}{2}}\odot\bar{\partial}\tilde{\mathcal{L}},\boldsymbol{\beta}^{-\frac{1}{2}}\odot\boldsymbol{v}\rangle$. By the homogeneity of $\tilde{q}_i$, we can further lower bound $\frac{\langle\bar{\partial}\tilde{\mathcal{L}},\boldsymbol{v}\rangle^2}{\rho^2}$ using $\tilde{\mathcal{L}}$. In other words, Lemma 2 ensures that the decreasing rate of the empirical loss $\tilde{\mathcal{L}}$ can be lower bounded by a function of itself. Based on the above methodology, we can prove that empirical loss will decrease to zero, while parameter norm will converge to infinity as the following lemma.

Lemma 4. *Let a function* $\boldsymbol{v}(t)$ *obey an adaptive gradient flow* $\mathcal{F}$ *with loss* $\tilde{\mathcal{L}}$ *and component learning rate* $\boldsymbol{\beta}(t)$, *where* $\tilde{\mathcal{L}}$ *satisfies Assumption 1. Then,* $\lim_{t\to\infty}\tilde{\mathcal{L}}(\boldsymbol{v}(t))=0$, *and consequently,* $\lim_{t\to\infty}\|\boldsymbol{v}(t)\|=\infty$.

5.3. Convergence to KKT point

We start by proving for any $t\geq t_1$, $\hat{\boldsymbol{v}}(t)=\frac{\boldsymbol{v}(t)}{\|\boldsymbol{v}(t)\|}$ is an approximate KKT point. Based on the surrogate margin that we construct in Section 5.1, we can further show for normalized $\boldsymbol{v}$ is an approximate KKT point as the following Lemma :

Lemma 5. *Let* $\hat{\boldsymbol{v}}$ *and* $\widehat{\bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v})}$ *be* $\boldsymbol{v}$ *and* $\bar{\partial}\tilde{\mathcal{L}}(\boldsymbol{v})$ *respectively normalized by their* L^2 *norms. Then* $\hat{\boldsymbol{v}}(t)$ *is a* $\left(\mathcal{O}(1-\langle\hat{\boldsymbol{v}}(t),\widehat{-\bar{\partial}\tilde{\mathcal{L}}(t)}\rangle),\mathcal{O}\left(\frac{1}{\log\frac{1}{\tilde{\mathcal{L}}(t)}}\right)\right)$ *KKT point of optimization problem* (P) *in Theorem 2.*

We made some explanations to Lemma 5: by the results in Section 5.2, we have $\lim_{t\to\infty}\mathcal{O}\left(\frac{1}{\log\frac{1}{\tilde{\mathcal{L}}(t)}}\right)=0$. Therefore, we only need to find a convergent series $\tilde{\boldsymbol{v}}(t)$ with $1+\langle\hat{\boldsymbol{v}}(t),\widehat{\bar{\partial}\tilde{\mathcal{L}}(t)}\rangle$ goes to zero.

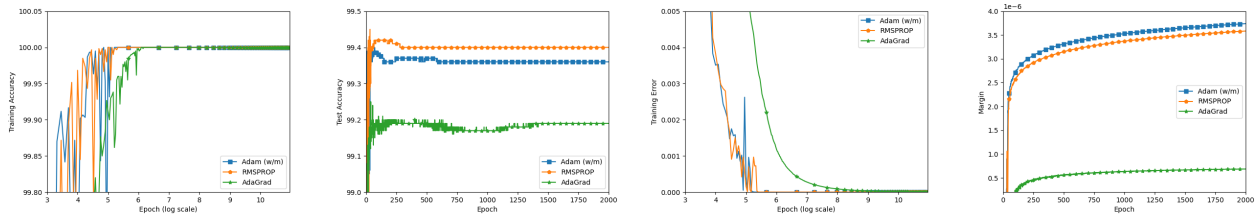
For this purpose, we construct an approximate norm $\tilde{\rho}(t)$ as $\sqrt{\rho(t)^2-2\int_{t_1}^t\left\langle\boldsymbol{v}(\tau),\boldsymbol{\beta}^{-\frac{1}{2}}(\tau)\odot\frac{d\boldsymbol{\beta}^{-\frac{1}{2}}}{\mathrm{d}t}(\tau)\odot\boldsymbol{v}(\tau)\right\rangle d\tau}$, which measures the increasing of $\boldsymbol{v}(t)$. $1-\langle\hat{\boldsymbol{v}}(t),\widehat{-\bar{\partial}\tilde{\mathcal{L}}(t)}\rangle$ can then be bound by the next lemma:

Lemma 6. *For any* $t_3>t_2\geq t_1$, *there exists a* $\xi\in[t_2,t_3]$, *such that*

$$\left(\langle\hat{v}(\xi),\widehat{-\bar{\partial}\tilde{\mathcal{L}}(\xi)}\rangle^{-2}-1\right)\leq\boldsymbol{o}\left(\frac{1}{\log\tilde{\rho}(t_3)-\log\tilde{\rho}(t_2)}\right),$$

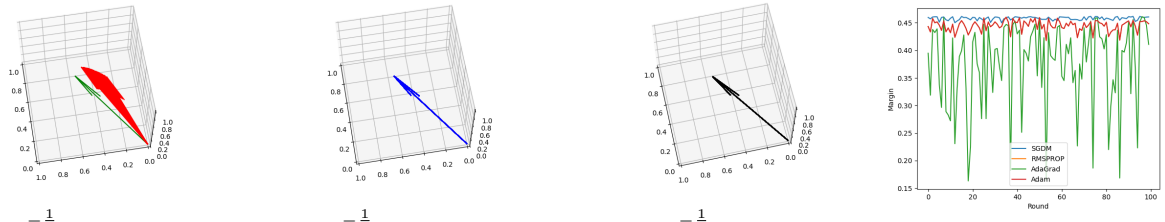
and

$$\|\hat{\boldsymbol{v}}(\xi)-\hat{\boldsymbol{v}}(t_2)\|\leq\mathcal{O}\left(\log\tilde{\rho}(t_3)-\log\tilde{\rho}(t_2)\right).$$



(a) Training Accuracy (b) Test Accuracy (c) Training Loss (d) Normalized Margin

Figure 1. Observation of normalized margin and generalization performance of different optimizers on MNIST. While all optimizers end with training accuracy 100% in (a), $1 -$ *test accuracy* can reflect the generalization error.



(a) $h_\infty^{-\frac{1}{2}}$ in AdaGrad (b) $h_\infty^{-\frac{1}{2}}$ in RMSProp (c) $h_\infty^{-\frac{1}{2}}$ in Adam (d) Margin

Figure 2. Direction of limit of conditioner in AdaGrad, RMSProp, and Adam (w/m) with different realizations of random initialization. In (a)-(c), the green vector stands for the isotropic direction $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. One red vector in (a) stands for direction of $(\boldsymbol{h}_\infty^A)^{-\frac{1}{2}}$ in one experiment. Blue vector in (b) stands for direction of $(\boldsymbol{h}_\infty^R)^{-\frac{1}{2}}$ under different initialization. Black vector in (c) stands for direction of $(\boldsymbol{h}_\infty^M)^{-\frac{1}{2}}$ under different initialization. In (d), final values of the margin for the four algorithms are plotted (Adam (w/m) coincides with RMSProp).

Therefore, given a sequence of parameter direction $\{\hat{\boldsymbol{v}}(t_i)\}_{i=1}^\infty$ with limit $\hat{\boldsymbol{v}}$, we can always construct another sequence $\{t_i'\}_{i=1}^\infty$ with $\left(1 - \langle \hat{v}(t_i'), -\widehat{\tilde{\partial}\tilde{\mathcal{L}}(t_i')}\rangle\right)$ and $\hat{\boldsymbol{v}}(t_i) - \hat{\boldsymbol{v}}(t_i')$ converging to zero.

Combining Lemma 5 and 6, for any convergent direction $\bar{\boldsymbol{v}}$, we can construct a series of $\{t_i\}_{i=1}^\infty$, such that $\hat{\boldsymbol{v}}(t_i)$ is $(\varepsilon_i, \delta_i)$ KKT point, with $\lim_{i\to\infty} \hat{\boldsymbol{v}}(t_i) = \bar{\boldsymbol{v}}$, and $\lim_{i\to\infty} \varepsilon_i = \lim_{i\to\infty} \delta_i = 0$. On the other hand, constraints of (P) satisfies Mangasarian-Fromovitz constraint qualification (see Appendix A.2), which ensures that $\bar{\boldsymbol{v}}$ is a KKT point of (P), and completes the proof.

6. Experiments

6.1. Observations on Normalized Margin and Generalization Performance

In this section, we conduct experiments to verify the theoretical results. We train a homogeneous neural networks using AdaGrad, RMSProp and Adam (w/m) respectively. We adopt the homogeneous 4-layer convolutional neural network used in (Madry et al., 2018) as our model and use MNIST (LeCun, 1998) as the dataset. We use default learning rate on PyTorch platform for all the algorithms and Adam (w/m) adopts the same learning rate as Adam. Because our theory is established for full batch gradient without randomness, we set minibatch size to be 1024 which is relatively large to mimic the full batch gradient. We put more details on the network structure and the settings of hyper-parameters in Appendix F.1, where we also add standard SGD (with momentum) and Adam to observe influence of momentum.

We plot training accuracy, testing accuracy and training loss in Figure 1a, 1b, and 1c. We also plot the value of the normalized margin during training in Figure 1d. We have the following observations: (1) The normalized margins of AdaGrad, RMSProp and Adam (w/m) are lower bounded and the final normalized margin of AdaGrad is the lowest. It is consistent with our theoretical results. (2) The training loss of AdaGrad, RMSProp and Adam (w/m) goes to zero and AdaGrad achieves the lower test accuracy (the worse generalization), which shows the superiority of conditioners in RMSProp and Adam (w/m) on generalization. (3) Although our theory does not include momentum version of the algorithms, the normalized margin of SGD and Adam are also lower bounded , which shows potential on extension of our theory to momentum version.

6.2. Observations on Convergent Direction

In this section, we observe the direction of $\boldsymbol{h}_\infty$ on a simple case to illustrate that $\boldsymbol{h}_\infty$ of AdaGrad is anistropic and sensitive to initialization. The model we use is expressed as $\Phi(\boldsymbol{x}, \boldsymbol{w}, v) = v\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$, where $x \in \mathbb{R}^2, \boldsymbol{w} \in \mathbb{R}^2$ and $v \in \mathbb{R}$ and $\sigma(x)$ is the Leaky ReLU activation function, i.e.,

$\sigma(x) = x$ for $x \geq 0$ and $\sigma(x) = \frac{x}{2}$ for $x < 0$.

We repeat AdaGrad, RMSProp and Adam (w/m) for 100 rounds with different random seeds of initialization. We plot $\boldsymbol{h}_\infty^{-\frac{1}{2}}$ for AdaGrad, RMSProp and Adam in Figure 2 (a), (b) and (c), respectively. We can observe that the $\boldsymbol{h}_\infty^{-\frac{1}{2}}$ in AdaGrad are different for 100 runs and $\boldsymbol{h}_\infty^{-\frac{1}{2}}$ in RMSProp and Adam (w/m) are coincide. It indicates that $\boldsymbol{h}_\infty^{-\frac{1}{2}}$ in AdaGrad is sensitive to initialization. We also plot the value of the margin for the three algorithms under different initialization in Figure 2(d). We can observe that the margin of AdaGrad fluctuates under different initialization, while that for RMSProp and Adam (w/m) are smoother. We further show the relation between $\boldsymbol{h}_\infty^{-\frac{1}{2}}$ and the convergent direction of parameters in Appendix F.3. These results indicate that the convergent direction of AdaGrad is sensitive to initialization, which may hurt its generalization.

7. Conclusion

In this paper, we study the convergent direction of both continuous and discrete cases of adaptive optimization algorithms on homogeneous deep neural networks. We prove that RMSProp and Adam (w/m) will converge to the KKT points of the L^2 max-margin problem, while AdaGrad does not. The main technical contribution of this paper is to propose a general framework for analyses of adaptive optimization algorithms' convergent direction. In future, we will study how optimization techniques such as momentum, weight decay and stochastic noise in optimization algorithm influence the convergent direction.

References

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pp. 8141–8150, 2019.

Bartlett, P. and Shawe-Taylor, J. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel methods—support vector learning*, pp. 43–54, 1999.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6241–6250, 2017.

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

Chen, J., Zhou, D., Tang, Y., Yang, Z., Cao, Y., and Gu, Q. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204, 2015.

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

Davis, D., Drusvyatskiy, D., Kakade, S., and Lee, J. D. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1): 119–154, 2020.

Deng, L., Hinton, G., and Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 8599–8603. IEEE, 2013.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *ICML*, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 9461–9471, 2018b.

Hinton, G., Srivastava, N., and Swersky, K. Neural networks for machine learning lecture 6a overview of mini–batch gradient descent. 2012.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2018.

Ji, Z. and Telgarsky, M. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pp. 1772–1798, 2019.

Ji, Z. and Telgarsky, M. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

Keskar, N. S. and Socher, R. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kurdyka, K. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pp. 769–783, 1998.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Li, Y., Fang, E. X., Xu, H., and Zhao, T. Implicit bias of gradient descent based adversarial training on separable data. In *International Conference on Learning Representations*, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Luo, L., Xiong, Y., Liu, Y., and Sun, X. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2018.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Nacson, M. S., Gunasekar, S., Lee, J., Srebro, N., and Soudry, D. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In *International Conference on Machine Learning*, pp. 4683–4692. PMLR, 2019a.

Nacson, M. S., Lee, J., Gunasekar, S., Savarese, P. H. P., Srebro, N., and Soudry, D. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3420–3428. PMLR, 2019b.

Nacson, M. S., Srebro, N., and Soudry, D. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3051–3059. PMLR, 2019c.

Neyshabur, B., Salakhutdinov, R. R., and Srebro, N. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

Qian, Q. and Qian, X. The implicit bias of adagrad on separable data. In *Advances in Neural Information Processing Systems*, pp. 7761–7769, 2019.

Reddi, S., Zaheer, M., Sachan, D., Kale, S., and Kumar, S. Adaptive methods for nonconvex optimization. In *Proceeding of 32nd Conference on Neural Information Processing Systems (NIPS 2018)*, 2018.

Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Stein, E. M. and Shakarchi, R. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press, 2009.

Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in neural information processing systems*, pp. 4148–4158, 2017.

Witten, I. H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques, (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., 2005.

Xu, T., Zhou, Y., Ji, K., and Liang, Y. When will gradient methods converge to max-margin classifier under relu models? *arXiv preprint arXiv:1806.04339*, 2018.

Young, T., Hazarika, D., Poria, S., and Cambria, E. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S. C. H., et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhuang, J., Tang, T., Ding, Y., Tatikonda, S. C., Dvornek, N., Papademetris, X., and Duncan, J. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in Neural Information Processing Systems*, 33, 2020.