A Method for Enhancing Generalization of Adam by Multiple Integrations

Long Jin*, **Han Nong, Liangming Chen, Zhenming Su**

School of Information Science and Engineering, Lanzhou University, Lanzhou, China
{jinlongsysu, nonghan0017, lmchen}@foxmail.com, suzhm@lzu.edu.cn

Abstract

The insufficient generalization of adaptive moment estimation (Adam) has hindered its broader application. Recent studies have shown that flat minima in loss landscapes are highly associated with improved generalization. Inspired by the filtering effect of integration operations on high-frequency signals, we propose multiple integral Adam (MIAdam), a novel optimizer that integrates a multiple integral term into Adam. This multiple integral term effectively filters out sharp minima encountered during optimization, guiding the optimizer towards flatter regions and thereby enhancing generalization capability. We provide a theoretical explanation for the improvement in generalization through the diffusion theory framework and analyze the impact of the multiple integral term on the optimizer's convergence. Experimental results demonstrate that MIAdam not only enhances generalization and robustness against label noise but also maintains the rapid convergence characteristic of Adam, outperforming Adam and its variants in state-of-the-art benchmarks.

Code — https://github.com/LongJin-lab/MIAdam
Extended version — https://arxiv.org/abs/2412.12473

Introduction

An appropriate optimizer is essential to train a deep neural network (DNN), as it directly affects the training convergence and performance of a model (Yao et al. 2021). The goal of optimizers is usually to minimize (or maximize) a certain objective function, typically a loss function, which measures the gap between the predictions and ground-truth values. As a traditional optimizer, stochastic gradient descent (SGD) is a commonly used optimizer for training DNNs (Deng et al. 2023). However, SGD suffers from certain limitations, such as the need to precisely tune the learning rate, the uniform scaling of gradients in all directions, and the risk of being trapped in saddle points (Johnson et al. 2020; Liu et al. 2021). In order to address these challenges, adaptive learning rate optimizers are developed, offering more nuanced control over learning rates and improved convergence in diverse training scenarios. Among them, adaptive moment estimation (Adam) (Kingma and Ba 2015) is currently one of the most popular adaptive learning optimizers for its rapid convergence and efficient handling of sparse gradients. The combination of first-order and second-order moments in Adam enables the effective incorporation of momentum-based optimization and adaptive learning rate methods, thereby enhancing its overall efficiency and applicability in various neural network training contexts. Despite being widely used, Adam also exhibits certain limitations, such as the inferior generalization capabilities compared to SGD in some scenarios (Wilson et al. 2017; Luo et al. 2019; Zou et al. 2021). Therefore, several enhanced variants of Adam are developed to alleviate the issue of poor generalization. Switching from Adam to SGD (SWATS) (Keskar and Socher 2017) is designed to start training with the Adam optimizer, then automatically switch to SGD, aiming to improve the model's generalization performance. However, this method can not maintain the original convergence rate of Adam to some extent. ND-Adam (normalized direction-preserving Adam) (Zhang 2018) meticulously preserves the direction of gradients for each parameter and produces the regularization effect akin to L2 weight decay. Despite this, its ability to enhance generalization is quite limited. AdaBound (Luo et al. 2019) employs dynamic constraints on

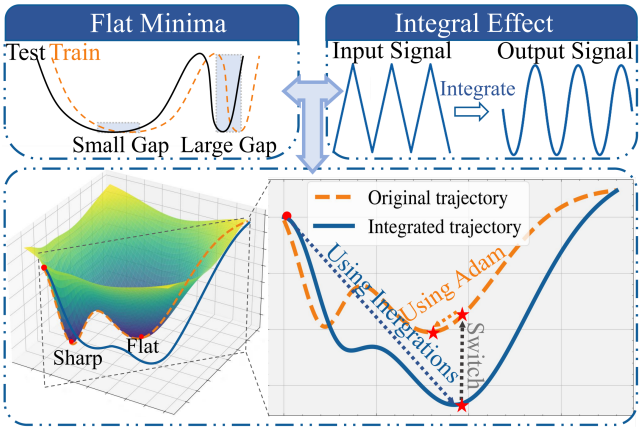


Figure 1: The idea of this work and the filtering effect of integrations on optimizer trajectories. The blue integrated trajectory represents an equivalent path that does not actually exist on the original loss landscape.

*Corresponding author.

the learning rate to achieve a smooth and gradual transition from adaptive methods to SGD, which enhances the generalization of a model and reduces the dependence on detailed learning rate adjustments. Nonetheless, a significant drawback of AdaBound is its potential for slow convergence in certain scenarios (Savarese 2019). Overall, these improved optimizers of Adam attempting to enhance the generalization of Adam are unable to simultaneously retain the rapid convergence characteristic of Adam and enhance generalization effectively.

Many studies show theoretically and empirically that the generalization performance of a model is highly correlated with its loss landscape in the parameter space (Hochreiter and Schmidhuber 1997; Chaudhari et al. 2019; Jiang et al. 2020; Petzka et al. 2021; Du et al. 2022). A crucial observation is that flat regions in the loss landscape tend to be associated with good generalization performance, while sharp or narrow regions may lead to overfitting (Mulayoff and Michaeli 2020; Sun et al. 2023). This means that optimizers can effectively improve the generalization of a model by converging to flat minima during the training process, which provides a new perspective to alleviate the issue of the poor generalization of Adam. In order to improve the generalization of a model by finding flat minima in the loss landscape, we propose multiple integral Adam (MIAdam), which is inspired by the effect that the multiple integral term can often serve as filters and noise suppressions in the field of signal processing and control systems (Roberts and Mullis 1987; Jin, Zhang, and Li 2015). MIAdam introduces a multiple integral term to the parameter update formula of Adam, utilizing the filtering effect of multiple integrations to smooth the optimizer's trajectory. As shown in Fig. 1, if we consider the trajectory as a time-varying input signal, integrating the signal is equivalent to filtering out the sharp minima encountered by the optimizer on the loss landscape, thereby enabling the optimizer to converge to flat minima. More details about the design of the MIAdam optimizer are discussed in Section MIAdam. The main contributions of this paper are summarized as follows.

- To the best of our knowledge, the method of introducing multiple integral term in the optimizer to find flat minima in the loss landscape is proposed for the first time. Furthermore, we propose a new optimizer based on Adam, which is called MIAdam.
- We provide theoretical analyses on MIAdam. Specifically, utilizing the diffusion theory framework in (Xie, Sato, and Sugiyama 2020), we prove that the multiple integral term enables MIAdam to generalize better than Adam under some assumptions. In addition, we also analyze the effect of multiple integral terms on convergence.
- The effectiveness of the proposed method is validated through image classification experiments, text classification experiments, and experiments that inject label noises into datasets. Experimental results demonstrate that MIAdam outperforms the Adam and its state-of-the-art (SOTA) variants on both generalization and robustness against label noises.

Preliminaries

In this section, core concepts about Adam and the related theoretical analyses on the relationship between flat minima and generalization are briefly given to set the stage for the detailed exposition of MIAdam that follows.

Overview of Adam

The training procedure for a DNN can be primarily characterized as an optimization problem, which is defined as follows:

$$\min_{\boldsymbol{\theta}} \frac{1}{|S|} \sum_{k=1}^{|S|} \mathscr{L}(\boldsymbol{x}_k, \boldsymbol{y}_k; \boldsymbol{\theta}), \tag{1}$$

where $\mathscr{L}(\boldsymbol{x}_k, \boldsymbol{y}_k; \boldsymbol{\theta})$ represents the loss function; $\boldsymbol{\theta}$ denotes the parameter of the model; $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are the input and its corresponding ground-truth label, respectively; S is a subset of D and D is the training dataset. In the early stages of deep learning development, SGD emerges as a prevailing optimizer, with its parameter update formula expressed as follows:

$$\theta_{t+1,i} = \theta_{t,i} - \alpha g_{t,i}, \tag{2}$$

where α represents the learning rate; $\theta_{t,i}$ represents the i-th dimension of the parameter at discrete time t; $g_{t,i}$ is the gradient with respect to the parameter $\theta_{t,i}$. The gradient is formally defined as

$$g_{t,i} = \frac{\partial L(\theta_{t,i})}{\partial \theta_{t,i}}, \tag{3}$$

where $L(\boldsymbol{\theta}) = (1/|S|) \sum_{k=1}^{|S|} \mathscr{L}(\boldsymbol{x}_k, \boldsymbol{y}_k; \boldsymbol{\theta})$. Momentum is a strategy used to expedite convergence of SGD towards minima and escape saddle points on the loss landscape (Qian 1999). It is computed by accumulating previous gradients into the current gradient. The parameter update formula of SGD with momentum (SGDM) is shown as

$$\begin{cases} m_{t,i} = \beta m_{t-1,i} + g_{t,i}, & \text{(4a)} \\ \theta_{t+1,i} = \theta_{t,i} - \alpha m_{t,i}, & \text{(4b)} \end{cases}$$

where $m_{t,i}$ denotes the momentum at t and β is the hyperparameter is used to trade off between the current gradient and the accumulation of historical gradients. Adam refines the momentum formulation in Eq. (4a) and introduces an adaptive learning rate achieved through the computation of the first-order and the second-order moments concerning current gradients. The first-order and second-order moments are calculated by the following expressions:

$$\begin{cases} m_{t,i} = \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}, & \text{(5a)} \\ v_{t,i} = \beta_2 v_{t-1,i} + (1-\beta_2) g_{t,i}^2, & \text{(5b)} \end{cases}$$

where $v_{t,i}$ denotes the second-order moment at t; the β_1 and β_2 are the exponential decay rates used to adjust the first-order and second-order moments, respectively. Furthermore, the parameter update formula of Adam is expressed as

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha \hat{m}_{t,i}}{\sqrt{\hat{v}_{t,i} + \epsilon}}, \tag{6}$$

where $\hat{m}_{t,i} = m_{t,i}/(1-\beta_1^t)$ and $\hat{v}_{t,i} = v_{t,i}/(1-\beta_2^t)$. The hyperparameter ϵ is assumed as a small value to prevent division by zero in the denominator.

The Relationship Between Flat Minima and Generalization

The generalization of DNNs has been extensively explored in recent years. In order to understand the phenomenon of generalization of DNNs, some of the existing research delves into understanding the relationship between loss landscapes and the generalization. A correlation between the flatness of the loss landscape and model generalization is revealed in (Hochreiter and Schmidhuber 1995). Subsequent investigations in (Hochreiter and Schmidhuber 1997) expand on this correlation and provide a method for identifying flat minima. In (Keskar et al. 2017), a definition of the sharpness of a specified point on a loss landscape is given. The study in (Dinh et al. 2017) introduces a reparameterization method and argues that previous sharpness measurements are inadequate for predicting generalization capabilities. Furthermore, it is demonstrated that the generalization capability is influenced by factors such as the batch size, higher-order "smoothness" terms characterized by the Lipschitz constant of the Hessian matrix, the loss function, and the number of parameters (Wang et al. 2018). Based on the above theoretical studies, empirical experiments extensively explore the intrinsic link between the generalization performance of a model and loss landscapes. A consensus emerging from these studies, including (Chaudhari et al. 2019; Jiang et al. 2020; Du et al. 2022; Petzka et al. 2021), is that flat minima usually yield better generalization compared to sharp minima.

MIAdam

In the parameter update formula of Adam, the first-order moment, as defined in Eq. (5a), is reformulated as follows (Kingma and Ba 2015):

$$m_{t,i} =(1-\beta_1)\sum_{j=0}^{t}\beta_1^{t-j}g_{j,i}. \tag{7}$$

Consequently, the parameter update formula of Adam is rewritten as

$$\theta_{t+1,i}=\theta_{t,i}-\frac{\alpha(1-\beta_1)\sum_{j=0}^{t}\beta_1^{t-j}g_{j,i}}{(1-\beta_1^t)\sqrt{\hat{v}_{t,i}+\epsilon}}. \tag{8}$$

When the learning rate α is sufficiently small, Eq. (8) is approximated in a continuous form as follows:

$$\mathrm{d}\theta_{\tilde{t},i}=\mu_{\tilde{t},i}\int_0^{\tilde{t}}\beta_1^{\tilde{t}-\tau}g_i(\tau)\mathrm{d}\tau, \tag{9}$$

where $\tilde{t}$ is the continuous time, $\mathrm{d}\tau$ is equivalent to α and $\mu_{\tilde{t},i}=-(1-\beta_1)/((1-\beta_1^{\tilde{t}})\sqrt{\hat{v}_{\tilde{t},i}+\epsilon})$. It is noteworthy that the integral term appears in Eq. (9). In signal processing, a continuous input signal $x(\tilde{t})$ that undergoes an integral operation is written as

$$y(\tilde{t})=\int_{\tilde{t}_0}^{\tilde{t}}x(\tau)\mathrm{d}\tau, \tag{10}$$

where $y(\tilde{t})$ is the integrated signal and the integration range is from $\tilde{t}_0$ to $\tilde{t}$. Ultimately, the resulting integral signal $y(\tilde{t})$ contains the cumulative information of the original signal $x(\tilde{t})$ at different points in time. After the integral operation, the high-frequency components of the signal are filtered out. Inspired by this, the trajectory of the optimizer on the loss landscape can be viewed as the input signal when training a DNN, and the sharp minima are equivalent to the high-frequency components in the signal. Integrating this signal is equivalent to filtering out the sharp minima encountered by an optimizer in the loss landscape, thereby guiding the optimizer toward convergence in flat regions. Therefore, to further achieve the effect of filtering out the sharp minima encountered by the optimizer, multiple integrations, an enhanced version of the integral operation, are introduced into the parameter update formula of Adam. Based on the process involving the integration as depicted in Eq. (10), we obtain the following equation:

$$\mathrm{d}\theta_{\tilde{t},i}=\overbrace{\mu_{\tilde{t},i}\int_0^{\tilde{t}}\kappa^{\tilde{t}-\tilde{t}_1}\int_0^{\tilde{t}_1}\cdots\int_0^{\tilde{t}_{n-2}}\kappa^{\tilde{t}_{n-2}-\tilde{t}_{n-1}}\int_0^{\tilde{t}_{n-1}}\beta_1^{\tilde{t}_{n-1}-\tilde{\tau}}g_i(\tilde{\tau})\mathrm{d}\tilde{\tau}\mathrm{d}\tilde{t}_{n-1}\cdots\mathrm{d}\tilde{t}_1}^{n\text{-th-order multiple integration}}, \tag{11}$$

where κ is multiple integration rate which adjusts the multiple integral term. Then, we perform cumulative operations on the first-order moments in the parameter update formula of Adam to transform the multiple integral term from a continuous form to its corresponding discrete form. Thus, the corresponding parameter update formula of Eq. (11) is derived as follows:

$$\begin{cases} m_{t,i}=\beta_1 m_{t-1,i}+(1-\beta_1)g_{t,i}, \\ \overline{m}_{t,i}^{(n)}=(1-\beta_1)\overbrace{\sum_{t_1=0}^{t}\kappa^{t-t_1}\sum_{t_2=0}^{t_1}\cdots\sum_{t_{n-1}=0}^{t_{n-2}}\kappa^{t_{n-2}-t_{n-1}}\sum_{t_n=0}^{t_{n-1}}\beta_1^{t_{n-1}-t_n}g_{t_n,i}}^{n\text{-th-order multiple summation}} \\ \qquad\quad =\sum_{t_1=0}^{t}\kappa^{t-t_1}\sum_{t_2=0}^{t_1}\cdots\sum_{t_{n-1}=0}^{t_{n-2}}\kappa^{t_{n-2}-t_{n-1}}m_{t_{n-1},i}, \\ \theta_{t+1,i}=\theta_{t,i}-\dfrac{\alpha^n\overline{m}_{t,i}^{(n)}}{(1-\beta_1^t)\sqrt{\hat{v}_t+\epsilon}}, \end{cases} \tag{12}$$

where the superscript $^{(n)}$ means the n-th-order multiple summation. According to the theoretical analyses in Section and the simulations in Fig. 2, although the multiple integral term helps an optimizer to find flat minima, the optimizer hovers around flat minima and does not converge. Thus, we only use the multiple integral term in the early stages of training, and after that, the optimizer switches to Adam to ensure that the training is convergent eventually. At this point, the multiple integral term is introduced into Adam, and this new optimizer is named MIAdam. The pseudo code for MIAdam is shown in Algorithm 1.

Note that the multiple integration is approximated by the multiple summation in Algorithm 1, which adds only n additional summation operations at each iteration for each dimension of the parameter. Therefore, MIAdam adds very little additional computational overhead compared to Adam. In the following text, we refer to Adam with an additional

Algorithm 1: MIAdam

Given: Learning rate: α;
exponential decay rates: β_1, β_2;
multiple integration rate: κ;
infinitesimal term: ϵ;
the order of the multiple integral item: n;
switching moment: ζ.
Initialize: Step time $t \leftarrow 0$;
first moment vector $m_{t=0,i} \leftarrow 0$;
second moment vector $v_{t,i} \leftarrow 0$, $\overline{m}^{(0)}_{t=0,i} \leftarrow m_{t=0,i}$.

while stopping criterion is not met **do**
  $t \leftarrow t+1$
  using Eq. (3) to get the gradient $g_{t,i}$
  $m_{t,i} \leftarrow \beta_1 m_{t-1,i} + (1-\beta_1) g_{t,i}$
  $v_{t,i} \leftarrow \beta_2 v_{t-1,i} + (1-\beta_2) g^2_{t,i}$
  if $t < \zeta$ **then**
    $m^{(0)}_{t,i} \leftarrow m_{t,i}$
    for $j = 1$ to n **do**
      $\overline{m}^{(j)}_{t,i} \leftarrow \kappa \overline{m}^{(j)}_{t-1,i} + \overline{m}^{(j-1)}_{t,i}$
    end for
    $\alpha_t \leftarrow \alpha^n$
    $\hat{m}_{t,i} \leftarrow \overline{m}^{(n)}_{t,i} / (1-\beta_1^t)$
  else
    $\alpha_t \leftarrow \alpha$
    $\hat{m}_{t,i} \leftarrow m_{t,i} / (1-\beta_1^t)$
  end if
  $\hat{v}_{t,i} \leftarrow v_{t,i} / (1-\beta_2^t)$
  $\theta_{t,i} \leftarrow \theta_{t-1,i} - \alpha_t \hat{m}_{t,i} / (\sqrt{\hat{v}_{t,i}} + \epsilon)$
end while

first-order integration as MIAdam1, and the one with an additional second-order integration as MIAdam2, and so on.

Generalization and Convergence Analyses

In this section, we present the theoretical analyses of the generalization and convergence associated with the addition of the multiple integral term to Adam, which does not involve the switching of optimizers. These analyses provide a theoretical foundation for our proposed optimizer.

Generalization Analyses

In this subsection, the diffusion theory framework is utilized to rigorously demonstrate that the incorporation of the multiple integral term enhances the generalization capabilities of the model. Specifically, generalization is quantitatively assessed by comparing the mean escape time, represented as ϕ, which indicates an optimizer's ability to escape from sharp minima. In the following analyses, we begin by delineating three fundamental assumptions that are crucial for the application of the diffusion theory framework (Xie, Sato, and Sugiyama 2020).

Assumption 1. The loss function around the critical point $\boldsymbol{p}$ is approximately written as

$$L(\boldsymbol{\theta}) = L(\boldsymbol{p}) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{p})^\top H(\boldsymbol{p})(\boldsymbol{\theta}-\boldsymbol{p}), \tag{13}$$

where the superscript $^\top$ means the transpose of a vector.

Assumption 2. (Quasi-equilibrium approximation). The system is in quasi-equilibrium near minima.

Assumption 3. (Low-temperature approximation). The system is under low temperature (small gradient noise).

Consequently, following the theoretical analyses in (Xie, Sato, and Sugiyama 2020; Xie et al. 2022), we can further deduce Theorem 1. The detailed proof is given in the Appendix.

Theorem 1. *Suppose that Assumption 1, Assumption 2, and Assumption 3 hold while saddle point $\boldsymbol{u}$ is the exit from sharp minimum $\boldsymbol{a}$. Then the mean escape time of MIAdam1 from sharp minimum $\boldsymbol{a}$ to flat minimum $\boldsymbol{b}$ through saddle point $\boldsymbol{u}$ before the switch is*

$$\begin{aligned}\phi_{\mathrm{MIAdam1}} = &\\ \pi &\left[\sqrt{1+\frac{4\alpha\sqrt{\mathcal{b}\,|H_{\boldsymbol{ue}}|}}{\tilde{t}(1-\beta_1)}}+1\right]\frac{\left|\det\left(H_{\boldsymbol{a}}^{-1}H_{\boldsymbol{u}}\right)\right|^{\frac{1}{4}}}{|H_{\boldsymbol{ue}}|}\\ &\exp\left[\frac{2\sqrt{\mathcal{b}}\Delta L}{\tilde{t}\alpha}\left(\frac{\varrho}{\sqrt{H_{\boldsymbol{ae}}}}+\frac{(1-\varrho)}{\sqrt{|H_{\boldsymbol{ue}}|}}\right)\right],\end{aligned} \tag{14}$$

where subscript $_{\boldsymbol{e}}$ denotes the escape direction; ϱ is the path-dependent parameter; $\mathcal{b} = |S|$ indicates the batch size; $\Delta L = L(\boldsymbol{u}) - L(\boldsymbol{a})$; H represents the Hessian matrix.

Comparing the mean escape time ϕ_{MIAdam1} obtained from Theorem 1 with that of Adam's in (Xie et al. 2022),

$$\begin{aligned}\phi_{\mathrm{Adam}} = &\pi\left[\sqrt{1+\frac{4\alpha\sqrt{\mathcal{b}\,|H_{\boldsymbol{ue}}|}}{(1-\beta_1)}}+1\right]\frac{\left|\det\left(H_{\boldsymbol{a}}^{-1}H_{\boldsymbol{u}}\right)\right|^{\frac{1}{4}}}{|H_{\boldsymbol{ue}}|}\\ &\exp\left[\frac{2\sqrt{\mathcal{b}}\Delta L}{\alpha}\left(\frac{\varrho}{\sqrt{H_{\boldsymbol{ae}}}}+\frac{(1-\varrho)}{\sqrt{|H_{\boldsymbol{ue}}|}}\right)\right],\end{aligned} \tag{15}$$

when $\tilde{t} > 1$, it is found that ϕ_{MIAdam1} is smaller than ϕ_{Adam}, indicating that Adam introduces an additional first-order integration which is more likely to escape from sharp minima and consequently converge to flat minima, thereby improving the generalization.

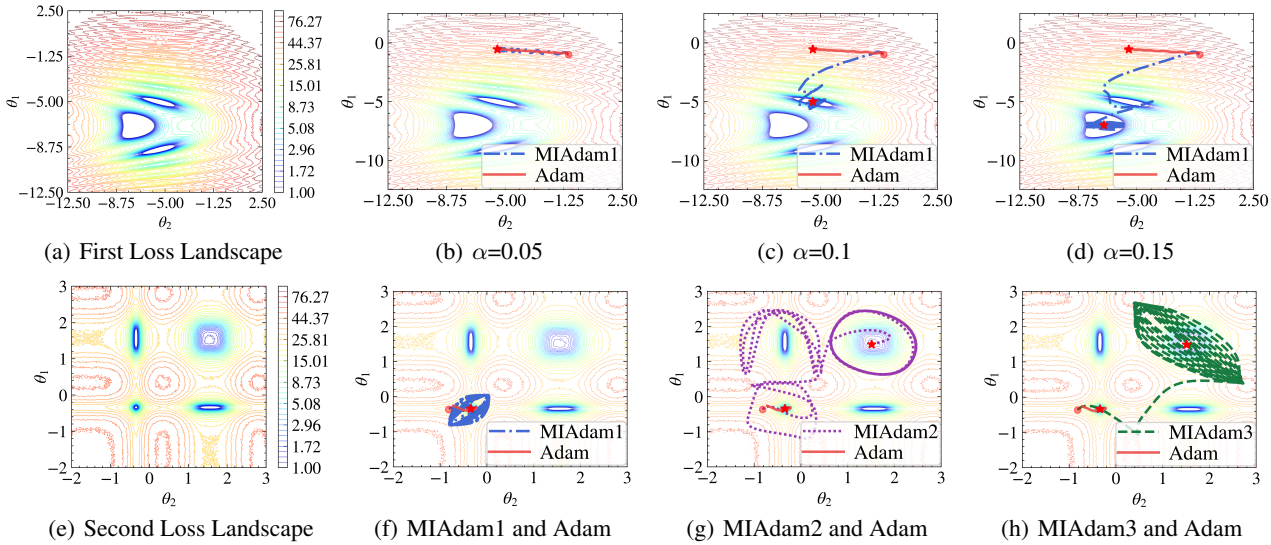
Convergence Analyses

In order to verify the effect of the multiple integral term on the convergence of the optimizer, we follow the analytical framework of Adam which is also used in this subsection. Concretely, the regret bound $R(\hat{t})$ is utilized to evaluate the convergence of the algorithm and is defined as follows:

$$R(\hat{t}) = \sum_{t=1}^{\hat{t}} f_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}} \sum_{t=1}^{\hat{t}} f_t(\boldsymbol{\theta}), \tag{16}$$

where $f_t(\cdot)$ is a convex loss function.

Theorem 2. *Assume that the convex function f_t has bounded gradients, $\|\nabla f_t(\boldsymbol{\theta})\|_2 \le \mathsf{g}$, $\|\nabla f_t(\boldsymbol{\theta})\|_\infty \le \mathsf{g}_\infty$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$ and distance between any $\boldsymbol{\theta}_t$ is guaranteed to be bounded, $\|\boldsymbol{\theta}_n - \boldsymbol{\theta}_m\|_2 \le \mathsf{d}$, $\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|_\infty \le \mathsf{d}_\infty$ for any*



(a) First Loss Landscape (b) α=0.05 (c) α=0.1 (d) α=0.15

(e) Second Loss Landscape (f) MIAdam1 and Adam (g) MIAdam2 and Adam (h) MIAdam3 and Adam

Figure 2: Simulations of trajectory of Adam and MIAdam on 2-parameter loss landscapes.

$m, n \in \{1, \ldots, \hat{t}\}$, and $\beta_1, \beta_2 \in [0, 1)$ satisfy $\beta_1^2/\sqrt{\beta_2} < 1$. Let $\alpha_t = \alpha/t^h$ and $\beta_{1,t} = \beta_1 \lambda^{t-1}, \kappa \in (0, 1], \lambda \in (0, 1)$. For the convex problem, the $R(\hat{t})$ of MIAdam1 before the switch satisfies

$$\lim_{\hat{t}\to\infty} \frac{R(\hat{t})}{\hat{t}} \neq 0. \tag{17}$$

The detailed proof of Theorem 2 is thoroughly presented in the Appendix. From Theorem 2, it is evident that merely adding an extra first-order integration to the parameter updating formula of Adam leads to the non-convergence of the optimizer. Although it is non-convergent, it effectively escapes sharp minima and hovers around flat minima in the loss landscape. This observation is corroborated by the simulation results shown in Figs. 2(f)-(h). As a result, the MIAdam's algorithm is structured to switch to Adam after a certain number of epochs to guarantee convergence.

Simulations and Experiments

In this section, we conduct the simulations on 2-parameter loss landscapes to illustrate the efficiency of MIAdam to escape from sharp minima. Furthermore, extensive empirical experiments are conducted to demonstrate that MIAdam outperforms Adam in terms of generalization and robustness against label noises.

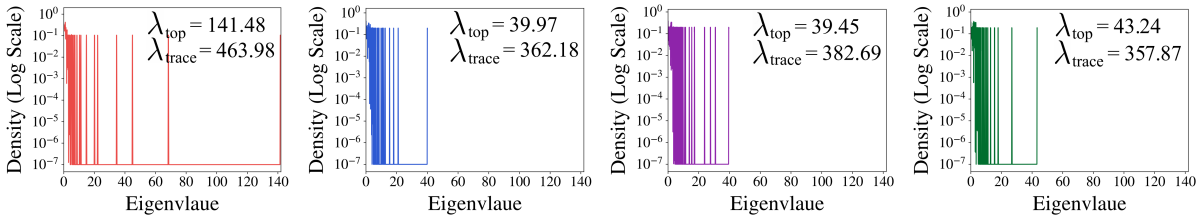
Simulations

This subsection mainly includes simulations demonstrating that MIAdam is easier to escape from a sharp minima compared to Adam and exploring the impact of learning rate on the optimization process. The first simulation is conducted on an elaborate 2-parameter loss landscape (Yang 2020) with one flat minima surrounded by two sharp minima, whose contour map is displayed in Fig. 2(a). On this loss landscape, the learning rates of Adam and MIAdam1 are respectively set to 0.05, 0.1, and 0.15, and the simulation results for their corresponding optimization trajectories are shown in Figs. 2(b)-(d). It is clear that MIAdam1 tends to escape from sharp minima and converge to flat minima compared to Adam on the 2-parameter loss landscape. Moreover, as the learning rate increases, MIAdam1 is able to converge to the flat minima. In contrast, Adam always shows poor convergence on this loss landscape and can not converge well to the flat minima or sharp minima. Therefore, our proposed method is effective in finding flat minima and can not be simply replaced by increasing the learning rate of Adam.

The second 2-parameter loss landscape used for simulations is depicted in Fig. 2(e), which contains a large number of sharp minima and flat minima. On this loss landscape, the optimization trajectories of MIAdam1, MIAdam2, MIAdam3, and Adam are compared in Fig. 2(f)-(h), respectively. Simulation results indicate that MIAdam1, MIAdam2, and MIAdam3 tend to converge toward flat minima, while the Adam optimizer tends to converge to the nearest sharp minima. It's worth noting that MIAdam3 exhibits more intense oscillations near the flat region compared to the MIAdam2. This suggests that increasing the order of multiple integration does not always lead to improved outcomes. Different starting points influence the trajectory of the optimizer. Therefore, to make the simulation results more convincing, we conduct additional simulations using 2,500 different starting coordinate points and calculate the sum of the absolute values of the eigenvalues of the Hessian matrix of the different optimizers for the final convergence points at different starting coordinate points and compare them. Due to space constraints, the simulation results are presented in the Appendix.

Experiments

The effectiveness of MIAdam is evaluated in this subsection through extensive empirical experiments. Initially, we con-



(a) Adam (b) MIAdam1 (c) MIAdam2 (d) MIAdam3

Figure 3: Comparisons of top Hessian eigenvalues λ_{top}, Hessian traces λ_{trace}, and full Hessian eigenvalue densities for loss landscapes on the CIFAR-10 dataset using ResNet18.

Optimizer	**ResNet18**				**ResNet50**			
	CIFAR-10(%)	**Time**	**CIFAR-100(%)**	**Time**	**CIFAR-10(%)**	**Time**	**CIFAR-100(%)**	**Time**
Adam	$93.89_{\pm 0.18}$	47m	$73.17_{\pm 0.26}$	47m	$93.69_{\pm 0.28}$	2h 45m	$75.24_{\pm 0.62}$	2h 47m
NAdam	$93.92_{\pm 0.11}$	46m	$73.08_{\pm 0.31}$	46m	$94.03_{\pm 0.07}$	2h 57m	$74.95_{\pm 0.08}$	2h 58m
AdamW	$93.74_{\pm 0.07}$	48m	$72.11_{\pm 0.23}$	47m	$93.80_{\pm 0.11}$	2h 45m	$74.40_{\pm 0.61}$	2h 48m
ND-Adam	$93.69_{\pm 0.08}$	44m	$72.26_{\pm 0.34}$	48m	$93.81_{\pm 0.18}$	2h 39m	$74.18_{\pm 0.35}$	2h 57m
Adamax	$93.68_{\pm 0.25}$	1h 28m	$74.27_{\pm 0.19}$	1h 27m	$94.10_{\pm 0.19}$	2h 48m	$76.30_{\pm 0.10}$	2h 52m
AdaBound	$92.93_{\pm 0.11}$	48m	$72.51_{\pm 0.23}$	49m	$93.14_{\pm 0.29}$	3h 1m	$73.87_{\pm 0.27}$	3h 2m
SWATS	$92.73_{\pm 0.10}$	1h 6m	$72.76_{\pm 0.27}$	54m	$92.38_{\pm 1.38}$	2h 51m	$72.83_{\pm 2.10}$	2h 50m
Adai	$93.86_{\pm 0.15}$	53m	$74.82_{\pm 0.09}$	53m	$93.95_{\pm 0.06}$	3h 32m	$76.07_{\pm 0.36}$	3h 18m
MIAdam1	$\mathbf{94.20_{\pm 0.12}}$*	47m	$\mathbf{75.03_{\pm 0.05}}$*	47m	$\mathbf{94.17_{\pm 0.10}}$	2h 49m	$\mathbf{76.96_{\pm 0.31}}$*	2h 47m
MIAdam2	$\mathbf{94.03_{\pm 0.12}}$	47m	$\mathbf{74.41_{\pm 0.38}}$	46m	$\mathbf{94.28_{\pm 0.15}}$*	2h 47m	$\mathbf{76.63_{\pm 0.15}}$	2h 46m
MIAdam3	$\mathbf{93.96_{\pm 0.17}}$	47m	$\mathbf{74.53_{\pm 0.22}}$	47m	$\mathbf{94.10_{\pm 0.10}}$	2h 48m	$\mathbf{76.51_{\pm 0.02}}$	2h 46m
	DenseNet121				**PyramidNet110**			
	CIFAR-10(%)	**Time**	**CIFAR-100(%)**	**Time**	**CIFAR-10(%)**	**Time**	**CIFAR-100(%)**	**Time**
Adam	$94.11_{\pm 0.01}$	3h 30m	$73.88_{\pm 0.34}$	3h 32m	$93.45_{\pm 0.10}$	2h 46m	$72.20_{\pm 0.09}$	2h 50m
NAdam	$94.39_{\pm 0.30}$	4h 23m	$65.71_{\pm 0.34}$	4h 24m	$93.33_{\pm 0.20}$	3h 34m	$71.26_{\pm 0.44}$	3h 32m
AdamW	$94.16_{\pm 0.11}$	3h 25m	$75.07_{\pm 0.15}$	3h 30m	$93.25_{\pm 0.08}$	2h 48m	$70.41_{\pm 0.69}$	2h 43m
ND-Adam	$94.11_{\pm 0.01}$	3h 23m	$74.59_{\pm 0.20}$	3h 34m	$93.00_{\pm 0.15}$	2h 58m	$70.72_{\pm 0.30}$	3h 11m
Adamax	$90.97_{\pm 0.15}$	3h 40m	$63.48_{\pm 0.08}$	3h 47m	$92.46_{\pm 0.24}$	4h 51m	$69.43_{\pm 0.28}$	5h 16m
AdaBound	$93.14_{\pm 0.10}$	4h 0m	$73.88_{\pm 0.35}$	4h 2m	$92.14_{\pm 0.22}$	3h 32m	$68.97_{\pm 0.39}$	3h 27m
SWATS	$93.64_{\pm 0.94}$	3h 35m	$75.62_{\pm 3.10}$	3h 45m	$89.42_{\pm 4.13}$	3h 35m	$49.82_{\pm 25.13}$	2h 29m
Adai	$94.45_{\pm 0.21}$	4h 28m	$76.77_{\pm 0.31}$	4h 42m	$93.50_{\pm 0.10}$	4h 18m	$71.94_{\pm 0.31}$	4h 5m
MIAdam1	$\mathbf{94.75_{\pm 0.10}}$*	3h 29m	$\mathbf{77.02_{\pm 0.10}}$*	3h 26m	$\mathbf{93.65_{\pm 0.08}}$*	2h 59m	$\mathbf{72.51_{\pm 0.24}}$*	2h 56m
MIAdam2	$\mathbf{94.43_{\pm 0.12}}$	3h 29m	$\mathbf{76.21_{\pm 0.33}}$	3h 32m	$93.02_{\pm 0.10}$	2h 52m	$71.96_{\pm 0.59}$	2h 52m
MIAdam3	$\mathbf{94.35_{\pm 0.03}}$	3h 30m	$\mathbf{76.54_{\pm 0.28}}$	3h 30m	$93.07_{\pm 0.25}$	2h 54m	$71.34_{\pm 0.17}$	2h 53m

Table 1: Top-1 test accuracy (mean±std) on CIFAR-10 and CIFAR-100.

duct image classification experiments with various neural network architectures on CIFAR[1] and ImageNet-1k[2], compared against widely-used adaptive learning rate optimizers, including Adam and its SOTA variants. Additionally, we utilize the fast computation method of Hessian information of loss landscapes provided in (Yao et al. 2020) for further comparative analyses. Subsequently, the effectiveness of the proposed MIAdam optimizer for text classification tasks is tested using the BERT and RoBERTa models across four distinct datasets (Lin et al. 2021). Finally, to validate the robustness against label noises of MIAdam, we perform image classification experiments on datasets injected with label noises. The results of MIAdam exceeding Adam are all bold, and the optimal experimental results are all marked by asterisks. Because of space constraints, the detailed experimental settings for all experiments are included in the Appendix.

Image Classification Experiments To enhance the conviction of our experimental results, we employ four different neural network architectures for image classification tasks on the CIFAR-10 and CIFAR-100 datasets: ResNet18 (He et al. 2016), ResNet50 (He et al. 2016), DenseNet121 (Huang et al. 2017), and PyramidNet110 (Han, Kim, and Kim 2017). For experiments on large-scale image datasets,

[1]http://www.cs.toronto.edu/ kriz/cifar.html
[2]https://www.image-net.org/

Optimizer	AlexNet(%)	ResNet18(%)	DenseNet121(%)
Adam	46.48	67.19	71.48
MIAdam1	**52.34***	**72.27***	**75.39***
MIAdam2	**49.61**	70.70	**74.22**
MIAdam3	43.36	**70.31**	**74.60**

Table 2: Top-1 test accuracies (mean±std) on ImageNet-1k

Dataset	Optimizer	BERT(%)	RoBERTa(%)
R8	Adam	$98.15_{\pm 0.02}$	$98.36_{\pm 0.05}$
	MIAdam1	$\mathbf{98.18_{\pm 0.03}}$*	$\mathbf{98.45_{\pm 0.05}}$*
	MIAdam2	$98.11_{\pm 0.10}$	$98.29_{\pm 0.10}$
	MIAdam3	$98.04_{\pm 0.09}$	$98.29_{\pm 0.06}$
R52	Adam	$96.36_{\pm 0.21}$	$96.21_{\pm 0.13}$
	MIAdam1	$\mathbf{96.42_{\pm 0.23}}$	$\mathbf{96.48_{\pm 0.01}}$
	MIAdam2	$\mathbf{96.57_{\pm 0.07}}$	$\mathbf{96.46_{\pm 0.23}}$
	MIAdam3	$\mathbf{96.65_{\pm 0.27}}$*	$\mathbf{96.65_{\pm 0.07}}$*
MR	Adam	$86.03_{\pm 0.30}$	$87.72_{\pm 0.09}$
	MIAdam1	$\mathbf{86.51_{\pm 0.09}}$*	$\mathbf{89.73_{\pm 0.17}}$*
	MIAdam2	$\mathbf{86.45_{\pm 0.19}}$	$\mathbf{89.54_{\pm 0.19}}$
	MIAdam3	$\mathbf{86.34_{\pm 0.09}}$	$\mathbf{89.54_{\pm 0.18}}$

Table 3: Test accuracies (mean±std) on text classification experiments

we utilize the AlexNet (Krizhevsky, Sutskever, and Hinton 2012), ResNet18, and DenseNet121 architectures for both training and testing on the ImageNet-1k. The classification performance of MIAdam is compared with optimizers such as Adam, NAdam (Dozat 2016), AdamW (Loshchilov and Hutter 2017), ND-Adam, Adamax (Kingma and Ba 2015), AdaBound, SWATS, and Adai (Xie et al. 2022). Detailed hyperparameters and experimental settings are presented in the Appendix. As observed from Table 1 and Table 2, MIAdam maintains a training time comparable to Adam while obtaining much better performance than Adam. To provide a comparison of the flatness in the final convergence regions, we compute top Hessian eigenvalues, Hessian traces, and full Hessian eigenvalue densities for loss landscapes of Adam, MIAdam1, MIAdam2, and MIAdam3 using DenseNet121 on the CIFAR-100 dataset in Fig. 3. Fig. 3 suggests that the multiple integral term is helpful in finding flatter minima in a specific neural network training task.

Text Classification Experiments We conduct text classification experiments by fine-tuning the pre-trained models, BERT and RoBERTa models, on three widely-used text datasets R8, R52, and Movie Review (MR). Each optimizer is run three times on each dataset using different network structures, with the mean and standard deviation of the test accuracy reported in Table 3. The experimental results indicate that MIAdam significantly outperforms Adam in text classification tasks.

Optimizer	Noise rate(%)			
	20	40	60	80
Adam	88.24	84.90	79.61	66.39
ND-Adam	87.52	84.10	78.34	63.51
AdaBound	86.51	82.46	76.58	57.86
SWATS	89.43	85.47	80.30	53.50
Adai	86.09	81.92	75.72	58.60
MIAdam1	**90.32***	**87.67***	**82.02***	**67.68***
MIAdam2	**89.13**	**85.03**	**79.84**	64.96
MIAdam3	**88.71**	**85.87**	79.40	64.27

Table 4: Top-1 test accuracy on CIFAR-10 under label noises.

Robustness Against Label Noises In this subsection, we investigate the capacity of MIAdam to withstand label noises in the training dataset, thereby validating its robustness against label noises. The ResNet18 network is trained by using Adam and MIAdam on the corrupted version of the CIAFR10 dataset, where some of its training labels are randomly flipped while the inputs are kept clean. The noise levels are 20%, 40%, 60%, and 80%. On each noise level, each optimizer is run only once. The remaining experimental settings are consistent with those used in the previous image classification experiments. As indicated in Table 4, MIAdam consistently achieves the highest test accuracy across all noise levels, underscoring MIAdam's superior robustness against label noises.

Conclusion

In this paper, we have proposed MIAdam, a new adaptive learning rate optimizer algorithm with a multiple integral term added to Adam. MIAdam smoothes the optimization trajectory through the filtering effect of the multiple integral term, enabling it to escape sharp local minima during training and converge towards flat minima, thereby alleviating the problem of poor generalization of Adam and improving the robustness against label noises while retaining the fast convergence of Adam. Utilizing the diffusion theory framework, we have provided the proof that incorporating the multiple integral term enhances the capability of the optimizer to escape sharp minima and converge to flatter minima, thus improving the generalization of the models. We have analyzed the convergence of MIAdam and provided a guarantee of convergence. The simulations have demonstrated that MIAdam is capable of finding flatter minima compared to Adam. For empirical analyses, We have conducted image classification experiments, text classification experiments, and experiments that inject label noises into datasets. The experimental results show that MIAdam has much better generalization and robustness against label noises than Adam. Future work will focus on introducing multiple integral terms into other optimizers.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62476115 and Grant 62176109, in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2023-ct05 and Grant lzujbky-2023-ey07, in part by the China Computer Federation (CCF)-Baidu Open Fund under Grant 202306, and in part by the Supercomputing Center of Lanzhou University.

References

Chaudhari, P.; et al. 2019. Entropy-SGD: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018.

Deng, X.; Sun, T.; Li, S.; and Li, D. 2023. Stability-based generalization analysis of the asynchronous decentralized SGD. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7340–7348.

Dinh, L.; Pascanu, R.; Bengio, S.; and Bengio, Y. 2017. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 1019–1028.

Dozat, T. 2016. Incorporating Nesterov momentum into Adam. In *International Conference on Machine Learning*.

Du, J.; Zhou, D.; Feng, J.; Tan, V.; and Zhou, J. T. 2022. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 23439–23451.

Han, D.; Kim, J.; and Kim, J. 2017. Deep pyramidal residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5927–5935.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hochreiter, S.; and Schmidhuber, J. 1995. Simplifying neural nets by discovering flat minima. *Advances in Neural Information Processing Systems*, 7: 529–536.

Hochreiter, S.; and Schmidhuber, J. 1997. Flat minima. *Neural Computation*, 9(1): 1–42.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jiang, Y.; Neyshabur, B.; Mobahi, H.; Krishnan, D.; and Bengio, S. 2020. Fantastic Generalization Measures and Where to Find Them. In *International Conference on Learning Representations*.

Jin, L.; Zhang, Y.; and Li, S. 2015. Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12): 2615–2627.

Johnson, T.; Agrawal, P.; Gu, H.; and Guestrin, C. 2020. AdaScale SGD: A user-friendly algorithm for distributed training. In *International Conference on Machine Learning*, 4911–4920.

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*.

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

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Lin, Y.; Meng, Y.; Sun, X.; Han, Q.; Kuang, K.; Li, J.; and Wu, F. 2021. BERTGCN: Transductive text classification by combining GCN and BERT. *arXiv preprint arXiv:2105.05727*.

Liu, Z.; Li, B.; Simon, J. B.; and Ueda, M. 2021. SGD can converge to local maxima. In *International Conference on Learning Representations*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Luo, L.; Xiong, Y.; Liu, Y.; and Sun, X. 2019. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*.

Mulayoff, R.; and Michaeli, T. 2020. Unique properties of flat minima in deep networks. In *International Conference on Machine Learning*, 7108–7118.

Petzka, H.; Kamp, M.; Adilova, L.; Sminchisescu, C.; and Boley, M. 2021. Relative flatness and generalization. *Advances in Neural Information Processing Systems*, 18420–18432.

Qian, N. 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1): 145–151.

Roberts, R. A.; and Mullis, C. T. 1987. *Digital signal processing*. Addison-Wesley Longman Publishing Co., Inc.

Savarese, P. 2019. On the convergence of AdaBound and its connection to SGD. *arXiv preprint arXiv:1908.04457*.

Sun, Y.; Shen, L.; Chen, S.; Ding, L.; and Tao, D. 2023. Dynamic Regularized Sharpness Aware Minimization in Federated Learning: Approaching Global Consistency and Smooth Landscape. In *International Conference on Machine Learning*.

Wang, H.; Keskar, N. S.; Xiong, C.; and Socher, R. 2018. Identifying generalization properties in neural networks. *arXiv preprint arXiv:1809.07402*.

Wilson, A. C.; Roelofs, R.; Stern, M.; Srebro, N.; and Recht, B. 2017. The marginal value of adaptive gradient methods in machine learning. *Advances in Neural Information Processing Systems*, 4148–4158.

Xie, Z.; Sato, I.; and Sugiyama, M. 2020. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Machine Learning*.

Xie, Z.; Wang, X.; Zhang, H.; Sato, I.; and Sugiyama, M. 2022. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International Conference on Machine Learning*, 24430–24459.

Yang, X. 2020. Stochastic gradient variance reduction by solving a filtering problem. *arXiv preprint arXiv:2012.12418*.

Yao, Z.; Gholami, A.; Keutzer, K.; and Mahoney, M. W. 2020. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, 581–590.

Yao, Z.; Gholami, A.; Shen, S.; Mustafa, M.; Keutzer, K.; and Mahoney, M. 2021. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, 10665–10673.

Zhang, Z. 2018. Improved Adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 1–2.

Zou, D.; Cao, Y.; Li, Y.; and Gu, Q. 2021. Understanding the generalization of Adam in learning neural networks with proper regularization. *arXiv preprint arXiv:2108.11371*.