Study on Loss Landscape Geometry for Improving Generalization in Adaptive Optimization Methods

Zubayr Khalid zukh00001@stud.uni-saarland.de
Enosh Peter Ponraj enpe00001@stud.uni-saarland.de
Sohom Mukherjee somu00003@stud.uni-saarland.de
Saarland University

***Abstract*—In this project we analyse the loss landscape geometry of adaptive optimization methods, using a measure of loss landscape curvature called sharpness. It is well known that flat minima are associated with better generalization. Improved methods that aid convergence of adaptive methods, namely ADAM, to flat minima are proposed, for improving its generalization. Our code is made publicly available at https://github.com/mukherjeesohom/OptML_Project.**

I. Introduction

One of the most popular improvements over SGD are algorithms with adaptive step-sizes such as ADAM [1]. However, it has been observed that SGD has better generalization performance than ADAM [2]. This has been attributed to convergence of ADAM to sharper minima in [3]. But this work does not provide any experimental results on the measure of loss landscape geometry. Another recent line of work have shown that sharp minima generalize poorly while flat minima generalize better [4], [5]. [6] proposed a practical algorithm algorithm named sharpness-aware minimization (SAM) to aid convergence of SGD to flat minima, and hence improve its generalization. We have made the following contributions in this work:

1) We extend extend the sharpness-aware minimization scheme to ADAM and empirically verify its improvement in generalization.
2) We derive a closed form expression for a measure of loss landscape curvature called *sharpness*, and conduct experiments to study sharpness for baseline as well as sharpness-aware optimization methods.

The rest of the paper is structured as follows. Section II introduces the notation and discusses the theory behind SAM and our contributions. The details of experimental setup, implementation and results are presented in Section III. Finally, we draw conclusions and insights from our experiments and discuss the scope of future work in Section IV.

II. Theory

A. Preliminaries

To begin, we set up the notations for developing the theory of sharpness-aware minimization. The training set is $\mathcal{S} := \bigcup_{i=1}^{n} \{(x_i, y_i)\}$ drawn i.i.d from the distribution $\mathcal{D}$. Consider a neural network parameterized by weights $w \in \mathcal{W} \subseteq \mathrm{R}^d$. We want to train the network to achieve low population loss $L_{\mathcal{D}}(w)$, which is approximated by minimizing the empirical or training loss $L_{\mathcal{S}}(w) = \frac{1}{n}\sum_{i=1}^{n} l(w, x_i, y_i)$, where l is the per data point loss function. For sharpness-aware optimization [6], instead of looking at the training loss alone, we look at the following loss function which promotes convergence to flat minima[1] by jointly minimizing the empirical loss and loss sharpness

$$\begin{aligned}\hat{w} &= \arg\min_{w} L_{\mathcal{S}}^{SAM}(w) \\ &= \arg\min_{w} \max_{||\epsilon||_2 \le \rho} L_{\mathcal{S}}(w+\epsilon) \\ &= \arg\min_{w} \max_{||\epsilon||_2 \le \rho} [L_{\mathcal{S}}(w+\epsilon) - L_{\mathcal{S}}(w)] + L_{\mathcal{S}}(w) \\ &= \arg\min_{w} J_{\mathcal{S}}(w) + L_{\mathcal{S}}(w)\end{aligned}$$

where $J_{\mathcal{S}}(w) := \max_{||\epsilon||_p \le \rho}[L_{\mathcal{S}}(w+\epsilon) - L_{\mathcal{S}}(w)]$ gives a formal definition of the *sharpness* of the minima, where a lower value corresponds to a flatter minima. ϵ is the weight perturbation vector, and ρ is the predefined constant radius of the ball defining the neighbourhood of w.

B. Sharpness Calculation

Proposition 1. *For a neural network parameterized by w and a predefined neighbourhood radius ρ, the sharpness of the loss landscape at a point w along the trajectory is given by*

$$J_{\mathcal{S}}(w) = \rho||\nabla_w L_{\mathcal{S}}(w)|| \tag{1}$$

Proof. It has been shown in [6] that using a first order Taylor approximation of the objective $L_{\mathcal{S}}$ around w, and for $p = 2$ norm, the value of ϵ that solves the inner maximization problem is given by

$$\hat{\epsilon} = \arg\max_{||\epsilon||_2 \le \rho} L_{\mathcal{S}}(w+\epsilon) = \rho \frac{\nabla_w L_{\mathcal{S}}(w)}{||\nabla_w L_{\mathcal{S}}(w)||} \tag{2}$$

From the definition of sharpness we have

$$\begin{aligned}J_{\mathcal{S}}(w) &= \max_{||\epsilon||_p \le \rho} [L_{\mathcal{S}}(w+\epsilon) - L_{\mathcal{S}}(w)] \\ &= L_{\mathcal{S}}(w+\hat{\epsilon}) - L_{\mathcal{S}}(w) \\ &= \hat{\epsilon}^T \nabla_w L_{\mathcal{S}}(w) \\ &= \rho \frac{\nabla_w L_{\mathcal{S}}(w)^T \nabla_w L_{\mathcal{S}}(w)}{||\nabla_w L_{\mathcal{S}}(w)||} \\ &= \rho ||\nabla_w L_{\mathcal{S}}(w)||\end{aligned}$$

[1] A flat minima is informally defined as a minimizer $\hat{w}$ in the neighbourhood of which training losses are low.

□

C. Algorithm

Finding $\hat{w}$ involves solving an inner maximization problem and outer minimization problem as explained in [6]. We extend this Algorithm by replacing the SGD in the outer minimization by a generic base optimizer (particularly we will focus on ADAM base optimizer in the next sections). Moreover, we also introduce the calculation of the closed form expression for sharpness that we obtained in Proposition 1 to our Algorithm 1.

Algorithm 1 SAM with Base Optimizer

Input: Training Loader $\mathcal{T}$, Loss Function l, Learning Rate η, Epochs E, Neighbourhood Radius ρ
Output: Trained model and sharpness array
Initialize $w = w_0$, $t = 0$, $\texttt{sharpness} = []$

for e in $\texttt{range}(E)$ **do**
  $\texttt{batch\_sharpness} = 0$
  for $(b, \mathcal{B})$ in $\texttt{enumerate}(\mathcal{T})$ **do**
    $\delta(w) = \nabla_w L_{\mathcal{B}}(w)$
    $\hat{\epsilon} = \rho \frac{\delta(w_t)}{||\delta(w_t)||}$
    $w_{\text{new}} = w_t + \hat{\epsilon}$
    $g_t = \delta(w_{\text{new}})$
    $w_{t+1} = \texttt{base\_optimizer}(w_t, g_t)$
    $\texttt{batch\_sharpness} += \rho||\delta(w_t)||$
    $t = t + 1$
  end for
  $\texttt{sharpness} = [\texttt{sharpness}; \frac{\texttt{batch\_sharpness}}{b}]$
end for
return $\hat{w}$, $\texttt{sharpness}$

III. Experiments

In all the following cases we study the loss landscape geometry and compare the generalization performance of a base optimizer (namely SGD and ADAM) with and without SAM. Our code is made publicly available at [2].

A. Experimental Setup

We evaluate our method on the image classification task. The CIFAR-10 dataset and ResNet18 [7], VGG16 [8] model architectures are used to present the results. The experiments use the same basic data augmentation techniques (horizontal flip, padding by four pixels, and random crop). Test accuracy is used as a metric for accessing the generalization performance of a model, and sharpness (Algorithm 1) is used as a measure of loss landscape geometry. To ensure a fair comparison we train all models for 200 epochs and batch size 128, with similar initialization. We have evaluated four different optimizer setups: SGD, SAM with SGD base optimizer, ADAM, and SAM with ADAM base optimizer. A learning rate of 0.1 is used for SGD and 0.001 is used for ADAM. We also employ a step decay type of learning rate scheduler, the details of which can be found in our code. All experiments are carried out for three different random seeds and the mean/standard deviation is reported.

[2]https://github.com/mukherjeesohom/OptML_Project

B. Baseline

For the baseline we train and obtain the generalization performance of the base optimizers (SGD and ADAM), without using SAM. In Table I, we can see the test accuracy for different settings of base optimizer/architecture .

C. Generalization Performance of SAM

The generalization performance of different models are recorded and plotted in Figure 1. From Table I, it can be seen that SAM improves the generalization performance across all the provided settings.

D. Loss Landscape Geometry of SAM

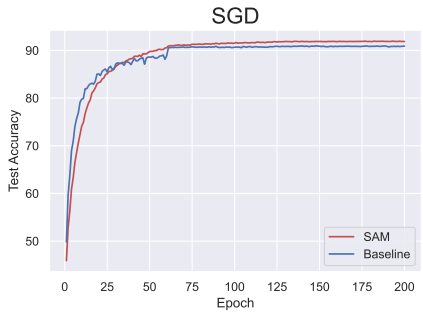
Figure 2 shows a plot of *per batch sharpness* (as calculated in Algorithm 1) versus epoch. It is clear that SAM has lower sharpness than the baseline for all settings, which is the expected result for sharpness-aware minimization. Moreover, the sharpness values also overall decreasing trend with epochs. However, it is remarkable that the sharpness decreases almost consistently for SGD while for ADAM it first increases and then decreases.

Optimizer	Architecture	
	ResNet18	VGG16
SGD	90.899 ± 0.137	90.089 ± 0.070
SGD + SAM	91.873 ± 0.102	90.453 ± 0.066
ADAM	92.716 ± 0.247	92.173 ± 0.310
ADAM + SAM	**94.896 ± 0.160**	**94.126 ± 0.112**

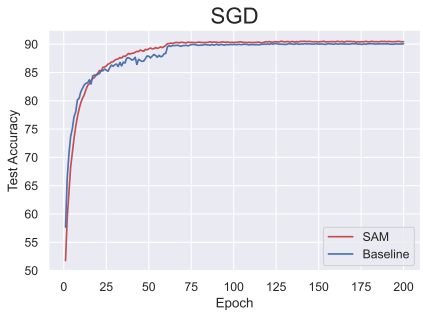
TABLE I: Comparison of generalization performance (test accuracy) for base optimizers (SGD and ADAM) with and without SAM.

IV. Conclusion

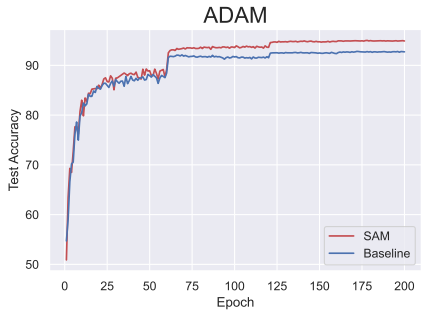
We show empirically that sharpness-aware minimization is equally effective in improving the generalization performance of ADAM, as it was shown for SGD in previous work [6]. By deriving the expression for sharpness and visualizing it as a function of epochs, we confirm that the performance gains are because of the decrease in sharpness of the loss landscape. However, it remains an open question as to why the sharpness values of SGD and ADAM show different trends. Moreover, all derivations concerning the theory of SAM, including the sharpness calculation involves a linear first order approximation of the training loss around w. This might not be fully accurate because the training loss is a highly nonlinear and non-convex function in general. While the current formulation of SAM involves sharpness calculation in an adversarial manner (where we find the ϵ in the neighbourhood of an optimizer that gives the worst training loss), other forms of calculation like average sharpness in a neighbourhood might also be interesting to explore in future work.



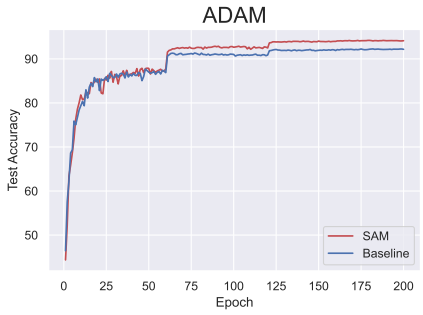
(a) ResNet18



(b) VGG16

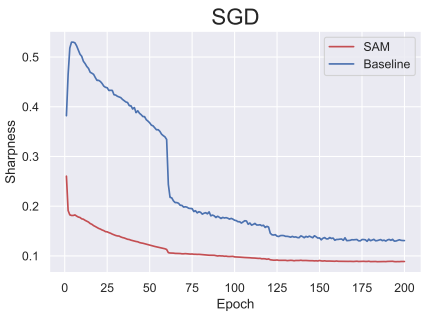


(c) ResNet18

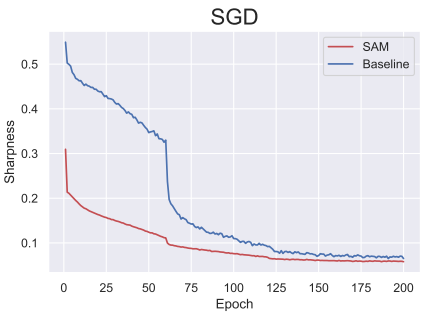


(d) VGG16

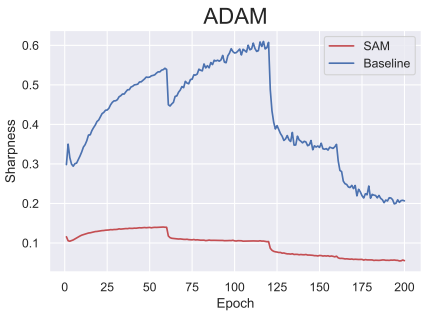
Fig. 1: Test accuracy plots for various setups.



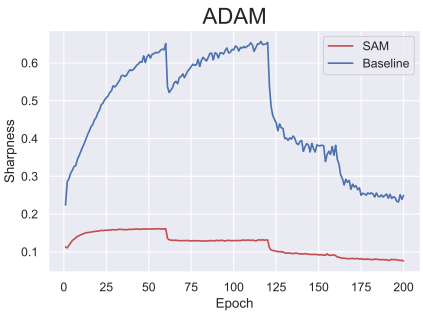
(a) ResNet18



(b) VGG16



(c) ResNet18



(d) VGG16

Fig. 2: Sharpness plots for various setups.

REFERENCES

[1] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[3] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi *et al.*, "Towards theoretically understanding why sgd generalizes better than adam in deep learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 285–21 296, 2020.

[4] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[5] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, "Fantastic generalization measures and where to find them," *arXiv preprint arXiv:1912.02178*, 2019.

[6] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," *arXiv preprint arXiv:2010.01412*, 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.