How Learning Rate, Batch Size, and Optimizer Influence Model Sharpness

Robin Patriarca[†], Agatha Duzan[†], Sylvain Mortgat[†]

[†]*École Polytechnique Fédérale de Lausanne (EPFL)*

***Abstract*—The correlation between the flatness of the loss landscape and model generalization remains a debated topic in machine learning. This study investigates how hyperparameters—batch size, learning rate, and optimizer choice—affect the sharpness of the loss landscape in a multilayer perceptron trained on CIFAR-10. Using average direction sharpness, we measure the model's sensitivity to parameter perturbations. Our findings reveal that sharpness is inversely related to batch size and learning rate for both SGD and AdaGrad optimizers. Specifically, smaller batch sizes and lower learning rates lead to sharper minima. These insights offer guidance for hyperparameter tuning to optimize both performance and generalization in deep learning models.**

I. Introduction

The flatness of the loss landscape is often considered to be positively correlated with model generalization. Keskar et al. [1] found that large batch training often leads to sharp minima, which is associated with performance degradation. This discovery has driven extensive theoretical and empirical research into the sharpness-generalization relationship. Techniques like Sharpness-Aware Minimization (SAM) [2] and Adaptive SAM (ASAM) [3] have shown practical improvements, particularly in vision transformers [4]. However, some sharpness-based methods, including SAM, face issues like sensitivity to model parameter re-scaling (scale-dependency) [5]. Recent works challenge the commonly held belief that flatness is always correlated with better generalization. For instance, Zhang et al. [6] found that the correlation between flatness and generalization can vary with different SGD variants, especially in image classification tasks like CIFAR-10. Andriushchenko et al. [7] argue that even reparametrization-invariant sharpness (like ASAM) is not a reliable indicator of generalization in modern settings.

Building on these findings, we propose to empirically explore the notion of average direction sharpness, as defined in [8]. Specifically, we study the sensitivity of sharpness in a multilayer perceptron to hyperparameters (batch size, learning rate) and optimizer (SGD, AdaGrad) on an image classification task.

In this work, we find that sharpness decreases with increasing batch size and increasing learning rate for both SGD and AdaGrad optimizers.

The paper is structured as follows: we first define sharpness. Next, we provide a detailed description of our models and training procedures. Finally, we present our empirical results and discuss their implications for model generalization.

II. Models and Methods

A. Average Direction Sharpness

In this study, we use the concept of average direction sharpness to quantify the flatness of the loss landscape. This approach offers a more stable and simpler estimation of sharpness compared to other methods that may require intricate calculations or adjustments for specific model architectures.

Sharpness around each model's parameters is assessed by measuring how much the loss increases when perturbing the model parameters by a small epsilon in various directions sampled randomly. This gives an empirical estimation of the local sharpness:

$$\text{Sharpness}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{g}\sim\mathcal{N}(0,I)}\left[L\left(\boldsymbol{\theta} + \varepsilon\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}\right)\right] - L(\boldsymbol{\theta}) \quad (1)$$

where L is the loss function, $\boldsymbol{\theta}$ represents the model parameters, ε is a small scalar, and $\boldsymbol{g}/\|\boldsymbol{g}\|_2$ is a unit vector indicating the direction of the perturbation.

We chose this definition of sharpness for various reasons:

- Stability: Average direction sharpness provides a robust measure that mitigates the effects of individual outliers in the perturbation directions. This stability is crucial when analyzing high-dimensional parameter spaces typical of neural networks.
- Simplicity: The method is straightforward to implement and does not require modifications to the training procedure or the introduction of additional hyperparameters.
- Relevance: Previous research, such as the work by Wen et al. [8], has shown that this measure correlates well with the model's generalization ability, making it a meaningful metric for our investigation.

By applying this method, we aim to analyze the relationships between hyperparameters and the loss landscape's geometry, and hopefully provide insights that can guide more effective hyperparameter tuning.

B. Model Description

To analyze sharpness with varying hyperparameters, we use a simple multi-layer perceptron (MLP) with three hidden layers of 512, 256, and 128 neurons, all using ReLU activations. The final linear layer maps to the 10 classes of the CIFAR-10 dataset [9]. Batch normalization is applied at the output to stabilize the learning process.

C. Training Setup & Sharpness Measurement

The models are trained using either SGD or AdaGrad as optimizers with various batch sizes and learning rates. Each model is characterized by a triplet (optimizer $\mathcal{O}$, batch size $\mathcal{B}$, learning rate γ). To ensure robust sharpness measurements, every triplet ($\mathcal{O}$, $\mathcal{B}$, γ) is initialized and trained n_{iter} times. We do not compute all possible combinations of $\mathcal{B}$ and γ. Instead, we vary $\mathcal{B}$ while keeping γ fixed at a baseline value and vice versa.

Training stops when the variation in loss between two consecutive epochs falls below a threshold τ for five consecutive epochs. This early stopping criterion ensures that the model has sufficiently converged, avoiding overfitting while ensuring that the sharpness measurement is meaningful.

Once the stopping criterion is met, sharpness is evaluated by sampling N vectors $\{\boldsymbol{g}_i\}_{i=1}^{N}$ from a multivariate standard normal distribution. The empirical average over these samples is used as a proxy for the expectation value in equation (1):

$$\mathbb{E}_{\boldsymbol{g}\sim\mathcal{N}(0,I)}\left[L\left(\boldsymbol{\theta}+\varepsilon\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}\right)\right]\approx\frac{1}{N}\sum_{i=1}^{N}L\left(\boldsymbol{\theta}+\varepsilon\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}\right). \qquad (2)$$

The sharpness measurements computed for each initialization are then averaged over the n_{iter} models. Table I lists the specific values for all the parameters. Note that batch size $\mathcal{B} = 50\,000$ corresponds to full gradient descent.

Parameter	**Value(s)**
Optimizer ($\mathcal{O}$)	SGD, AdaGrad
Batch size ($\mathcal{B}$)	$\{2^i\}_{i=5}^{15} \cup \{50\,000\}$
Baseline $\mathcal{B}$	2^7
Learning rate (γ)	$\{10^{-i}\}_{i=1}^{4} \cup \{5 \cdot 10^{-i}\}_{i=2}^{4}$
Baseline γ	10^{-4}
Number of iterations (n_{iter})	6
Stopping threshold (τ)	10^{-4}
Number of samples (N)	294
Perturbation radius (ε)	5

TABLE I
HYPERPARAMETERS VALUES USED IN THE TRAINING SETUP AND SHARPNESS MEASUREMENT. **(TODO: RECHECK THE VALUES)**

D. Reproducibility & Preprocessing

In practice, each of the n_{iter} initializations of a model ($\mathcal{O}$, $\mathcal{B}$, γ) is associated with different NumPy and PyTorch random seeds. Models with different iteration numbers $n \in \{1, 2, \ldots, n_{\text{iter}}\}$ are trained in parallel. The NumPy seed is used when sampling vectors $\boldsymbol{g}_i \sim \mathcal{N}(0, I)$ to compute sharpness using equation (2). The PyTorch seed, uniquely determined for each model instance via a SHA-1 hash of its identifier, is used to initialize weights and biases. Before training, all CIFAR-10 images are centered and normalized by subtracting the mean and dividing by the standard deviation of the dataset.

III. Results

Our experiments aimed to explore how varying batch sizes, learning rates, and optimizers influence the sharpness of the loss landscape in a multilayer perceptron trained on CIFAR-10. Sharpness was measured using the average direction sharpness metric, and 95% confidence intervals were constructed to quantify the statistical reliability of the averaged values.

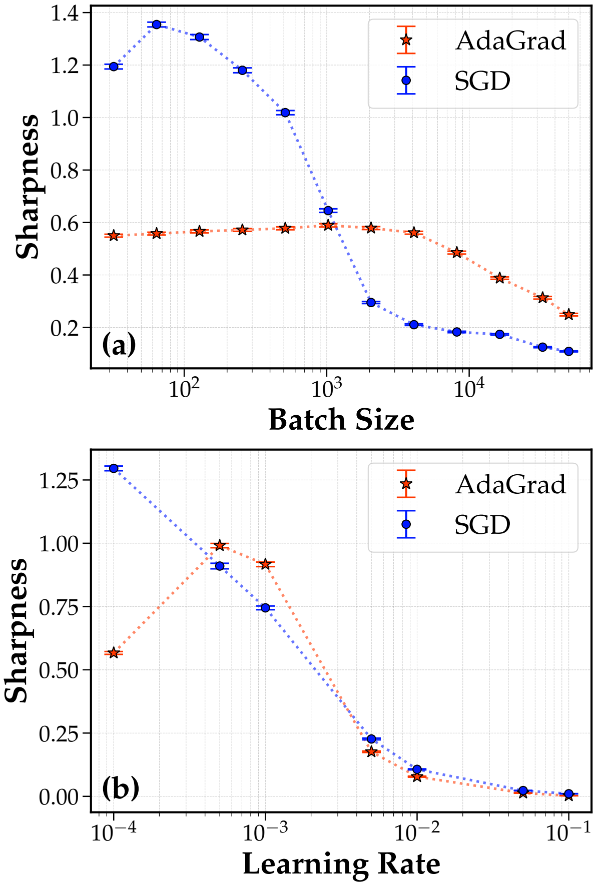

Fig. 1. Average direction sharpness against different (a) batch sizes with fixed $\gamma = 10^{-4}$, (b) learning rates with fixed $\mathcal{B} = 2^7$, for SGD and AdaGrad optimizers. Each measurement is an average over $N = 294$ randomly drawn samples. The error bars indicate the 95% confidence intervals.

A. Effect of Batch Size on Sharpness

Figure 1(a) shows the relationship between batch size and sharpness for both SGD and AdaGrad optimizers, keeping the learning rate fixed at the baseline $\gamma = 10^{-4}$. The results indicate a distinct trend:

- SGD's sharpness is inversely correlated to batch size, with a trend of decreasing sharpness as batch size increases.
- For AdaGrad, sharpness remains relatively stable across different batch sizes, showing only a slight decrease as the batch size increases.
- For large batch sizes, SGD outperforms AdaGrad in terms of finding flatter minima, while AdaGrad is better at finding flatter minima at smaller batch sizes.

B. Effect of Learning Rate on Sharpness

Figure 1(b) shows the impact of varying learning rates on sharpness while keeping the batch size fixed at the baseline $\mathcal{B} = 2^7$. Key observations include:

- Both AdaGrad and SGD benefit from increased learning rates in terms of finding flatter minima.
- At higher learning rates, both algorithms converge to similarly flat minima.
- For AdaGrad, sharpness also declines for the smallest learning rate ($\gamma = 10^{-4}$). On the other hand, SGD exhibits a more gradual reduction in sharpness as the learning rate increases.

Overall, these findings show how batch size, learning rate, and optimizer choices can influence the sharpness of the loss landscape. Such insights can help with designing training regimes that optimize both performance and generalization in deep learning models.

IV. Discussion

This section will discuss the implications of our results, their relevance to existing literature, and potential directions for future research.

A. Implications of Batch Size and Learning Rate on Sharpness

The observed inverse relationship between batch size and sharpness aligns with previous findings by Keskar et al. [1], which suggest that larger batch sizes tend to lead to flatter minima, thus potentially improving model generalization. Our results extend this understanding by quantifying this relationship using the average direction sharpness metric. Smaller batch sizes, associated with sharper minima, could imply more sensitive regions in the loss landscape, possibly leading to overfitting if not properly managed.

Similarly, the decrease in sharpness with increasing learning rates supports the theoretical foundations suggesting that higher learning rates can facilitate escape from sharp, narrow minima and guide the optimization process toward flatter regions. This trend was consistent across both optimizers used in our study, indicating a robust phenomenon that can be leveraged to tune hyperparameters for better model performance and generalization.

B. Optimizer-Specific Observations

Our findings show distinct behaviors for SGD and AdaGrad with respect to sharpness. While AdaGrad exhibits a more stable sharpness across different batch sizes, SGD demonstrates a significant sensitivity, with sharpness decreasing markedly as batch size increases. This could be attributed to AdaGrad's adaptive nature, which adjusts the learning rate dynamically, thus providing a more consistent trajectory in the parameter space.

The decline in sharpness for very small learning rates in AdaGrad suggests that extremely low learning rates may lead to overly cautious updates, trapping the optimization process in sharper regions. In contrast, SGD's gradual reduction in sharpness with increasing learning rates highlights its potential for more stable exploration of the loss landscape.

C. Limitations and Future Work

One limitation of our study is the focus on a relatively simple MLP architecture. Future research could extend this analysis to more complex models such as convolutional neural networks (CNNs) or transformers, which may exhibit different sharpness behaviors. Additionally, exploring alternative definitions of sharpness, including those invariant to parameter rescaling, could provide a more comprehensive understanding of the loss landscape's geometry.

Moreover, investigating the interaction between other hyperparameters, such as momentum or decay rates in adaptive optimizers, could provide deeper insights. On top of that, the impact of regularization techniques on sharpness and generalization should also be explored to develop more holistic hyperparameter tuning strategies.

Finally, it should be noted that sharpness alone is not always a sufficient descriptor of local minima, as highlighted by He et al. in their study on ”Asymmetric Valleys: Beyond Sharp and Flat Local Minima” [10]. They argue that some minima are asymmetric: flat in most directions but sharp in a few, which makes sharpness less relevant to describe the loss landscape.

Additionally, the work by Andriushchenko et al. [7] challenges the traditional view by showing that sharpness does not always correlate well with generalization. They observed cases where sharper minima could generalize better, which contradicts the conventional wisdom that flatter minima are always preferable. This suggests that the relationship between sharpness and generalization is highly dependent on the specific setup and hyperparameter configurations.

Therefore, while sharpness can be a useful metric for understanding and improving model generalization, it should not be used alone.

V. Summary

This study contributes to a deeper understanding of the optimization landscape in deep learning by empirically analyzing how different training strategies affect the sharpness of local minima. The insights from this analysis can be further used to tune hyperparameters not just for performance on the training set but also for improved generalization. Future work might explore more complex models or alternative definitions of sharpness that could provide additional insights into the geometry of high-dimensional loss landscapes encountered in practice.

References

[1] 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,” *CoRR*, vol. abs/1609.04836, 2016. [Online]. Available: http://arxiv.org/abs/1609.04836

[2] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” *ArXiv*, vol. abs/2010.01412, 2020. [Online]. Available: https://arxiv.org/abs/2010.01412

[3] J. Kwon, J. Kim, H. Park, and I. Choi, “Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks,” *ArXiv*, vol. abs/2102.11600, 2021. [Online]. Available: https://arxiv.org/abs/2102.11600

[4] X. Chen, C.-J. Hsieh, and B. Gong, “When vision transformers outperform resnets without pretraining or strong data augmentations,” *ArXiv*, vol. abs/2106.01548, 2021. [Online]. Available: https://arxiv.org/abs/2106.01548

[5] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, “Sharp minima can generalize for deep nets,” in *International Conference on Machine Learning*, 2017. [Online]. Available: https://arxiv.org/abs/1703.04933

[6] S. Zhang, I. Reid, G. V. P’erez, and A. A. Louis, “Why flatness does and does not correlate with generalization for deep neural networks,” 2021. [Online]. Available: https://arxiv.org/abs/2103.06219

[7] M. Andriushchenko, F. Croce, M. Müller, M. Hein, and N. Flammarion, “A modern look at the relationship between sharpness and generalization,” *ArXiv*, vol. abs/2302.07011, 2023. [Online]. Available: https://arxiv.org/abs/2302.07011

[8] K. Wen, T. Ma, and Z. Li, “How does sharpness-aware minimization minimize sharpness?” *ArXiv*, vol. abs/2211.05729, 2022. [Online]. Available: https://arxiv.org/abs/2211.05729

[9] A. Krizhevsky and G. Hinton, “Cifar-10 (canadian institute for advanced research),” 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[10] H. He, G. Huang, and Y. Yuan, “Asymmetric valleys: Beyond sharp and flat local minima,” *CoRR*, vol. abs/1902.00744, 2019. [Online]. Available: http://arxiv.org/abs/1902.00744

Appendix