Generalization properties of learning algorithms and Sharpness Aware Minimization based on Minimum Sharpness

Irina Bejan
irina.bejan@epfl.ch

Miguel-Angel Sanchez Ndoye
miguel-angel.sanchezndoye@epfl.ch

Jana Vuckovic
jana.vuckovic@epfl.ch

***Abstract*—One of the biggest challenges in deep learning is the understanding of generalization. Sharpness is one of the indicators of generalization properties, that performs well in practice, but is not yet understood. Sharpness aware minimization (SAM) is a new state-of-the-art technique based on simultaneously minimizing both the loss and the sharpness. In this paper, we investigate a recently introduced notion of sharpness known as minimum sharpness. We investigate its correlation with the generalization gap by considering many different optimizers and SAM. Finally, we tackle the question of adaptivity of learning algorithms, as that also has an impact on generalization, and investigate how the choice of optimizer influences the sharpness.**

Keywords: generalization, sharpness aware minimization, minimum sharpness, (non-)adaptive learning algorithms

I. Introduction

Generalization is a crucial and yet still unresolved question in the field of Deep Learning. The ability of a model to generalize well is influenced by a lot of different factors. For example, it is known that different optimizers can lead to very different solutions. Particularly, [1] gave an example in which non-adaptive learning algorithms, such as Stochastic Gradient Descent (SGD) and Polyak Heavy Ball (PHB), generalize better than adaptive ones, such as Adam. On the other hand, [2] argues that the performance of different learning algorithms varies across tasks.

Another aspect that has an impact on generalization is the shape of the loss landscape. It has been observed that flat minima tend to generalize better [3]. However, [4] argues that sharp minima, considering that sharpness is measured as a trace of a Hessian of the loss, can also generalize well. It points out that the reason for this is that trace of the Hessian of the loss is not invariant to α-scale-transformations, while the performance of a neural network is. For that reason, new notions of sharpness have been introduced, such as normalized sharpness [5], minimum sharpness [6], sharpness based on PAC-Bayesian theory [7], etc. An empirical study that shows the success of sharpness-based measures is presented in [8]. Moreover, [9] proposes a new minimization procedure, based on controlling the sharpness, known as Sharpness Aware Minimization (SAM).

In this paper, we want to tackle both questions of learning algorithms and sharpness. For this reason we consider 6 different optimizers - SGD, PHB, Adam [10], Adagrad [11], AdaShift [12] and AdaBound [13]; both alone and as background optimizers of SAM. Firstly, we show that SAM generalizes better and converges to less sharp minima, using the notion of minimum sharpness, introduced in [6]. We chose this notion because it is invariant to α-scale-transformations, not computationally heavy, and yet not investigated much. Then, we investigate whether there is a correlation between minimum sharpness and the generalization gap. Finally, inspired by [1], we compare adaptive and non-adaptive algorithms and comment on whether the minima were found to differ concerning the minimum sharpness.

The paper is organized as follows. Firstly, in section II, we explain the concepts used in the rest of the paper - SAM and minimum sharpness. Then, in section III, we explain our experiments in detail and present our results. Finally, we conclude in section IV.

II. Sharpness

A. Sharpness Aware Minimization (SAM)

As stated in part I, it has been observed that flat minima tend to generalize better. As a consequence it might be beneficial for generalization not only to minimize the loss, but also to minimize the sharpness of the local minimum reached. This is what SAM, proposed in [9], seeks to do, by minimizing the following objective function:

$$\min_{\boldsymbol{w}} L_S^{SAM}(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2 \tag{1}$$

where

$$L_S^{SAM}(\boldsymbol{w}) = \max_{\|\varepsilon\|_p \leq \rho} L_S(\boldsymbol{w} + \boldsymbol{\varepsilon}) \tag{2}$$

for some $\rho \geq 0, p \geq 1$. Combining (1) and 2 we obtain that the objective that is being minimized can be rewritten as

$$\left(\max_{\|\varepsilon\|_p \leq \rho} L_S(\boldsymbol{w} + \boldsymbol{\varepsilon}) - L_S(\boldsymbol{w}) \right) + L_S(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2. \tag{3}$$

This is important because the two first terms represents the sharpness while the two last terms consist in a regular loss function with regularization. In the paper, it is claimed that

$$\nabla_{\mathbf{w}} L_S^{SAM}(\boldsymbol{w}) \approx \nabla_{\mathbf{w}} L_S(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\varepsilon}}} \tag{4}$$

where $\hat{\boldsymbol{\varepsilon}}$ is a vector that can easily be computed using $\nabla_{\mathbf{w}} L_S(\boldsymbol{w})$. As a consequence, the SAM optimization scheme consists in iteratively using a standard optimizer, also known as background optimizer, such as SGD or Adam, to minimize $L_S^{SAM}(\boldsymbol{w})$ using (4).

It has been shown empirically in [9] that using SAM can improve generalization performance on a variety of tasks including image classification, finetuning pretrained models, as well as learning with noisy labels. In this paper, we use SAM with different background optimizers, while in the literature it is usually used with SGD. The code used is available on their Github repository[1].

B. Minimum sharpness

One naive way of calculating the sharpness is as the trace of the Hessian of the loss function. However, this is not invariant to α-scale-transformations, and consequently does not show very high correlation with the generalization gap [4]. α-scale transformations are a particular class of transformations of the parameters of the

[1] https://github.com/davda54/sam

model that do not change the function induced by the model. Such a transformation depends on D scalars $\alpha_1, ...\alpha_D$, such that $\prod_{d=1}^{D} \alpha_d = 1$, where D is the number of layers of the model. It transforms the parameters by multiplying the parameters of each layer with the corresponding scalar. Another difficulty with this approach is the fact that the computation of the Hessian is quite heavy. Minimum sharpness is a notion of sharpness introduced in [6]. It overcomes the two previous problems by introducing an efficient way of approximating the Hessian and defining the minimum sharpness as

$$MS_w = \min_{\alpha} \mathrm{Tr}[H_{\alpha(w)}], \tag{5}$$

where α represents any α-scale transformation. Thus, by definition, minimum sharpness is invariant to α-scale-transformations.

In [6], it is shown that minimum sharpness correlates with the generalization gap, both in the case of convolutional and fully connected neural networks. Their experiments were done on the MNIST dataset, in the synthetically made regime, in which they controlled the generalization gap by shuffling labels. Due to the shuffling of labels, they were able to vary the generalization gap from 0 to 1. We use minimum sharpness differently. Firstly, our models contain both convolutional and linear layers. Secondly, we use different datasets. Finally, we measure the correlation between the generalization gap and the sharpness in non-artificial setting, meaning that the generalization gaps obtained by our experiments are relatively similar. The code we used is taken from their github repository[2].

III. Experiments and results

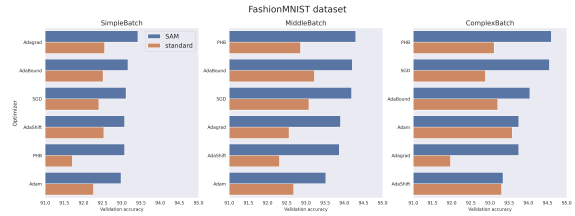
A. Experiments in detail

Our experiments are performed on the FashionMNIST and CIFAR-10 datasets. We consider three different architectures. Each architecture consists of several convolutional layers, with ReLU activation functions, followed by 2 linear layers. As suggested in [8], we use batch normalization in order to ensure that the training loss decreases sufficiently. The three architectures used are relatively similar, but differ in the number of convolutional layers - 4, 6 and 9, so we call those architectures respectively *SimpleBatch*, *MiddleBatch* and *ComplexBatch*, as their complexity increases. Also, we comment that there are minor differences in a single architecture for different datasets, due to different shapes of images and differences in the number of channels of FashionMNIST and CIFAR-10. A summary of the architectures is given, for FashionMNIST, in the figures 5, 6, 7 (Appendix). We use the Cross Entropy Loss and a batch size is set to 128. We consider 6 different learning algorithms: SGD, PHB, Adam, Adagrad, AdaBound[3] and AdaShift [4], used both alone and as a background optimizer of SAM. The learning rate used for SGD is 0.1. For PHB, we set the learning rate to be 0.1, or 0.01 if the learning rate of 0.1 does not lead to convergence; and the momentum coefficient to be 0.8. For adaptive algorithms we use the default learning rates, as suggested by [2], which states that the evaluation of multiple optimizers with default learning rates leads to similar results as would be obtained by tuning the learning rates of each optimizer. For each architecture, dataset and optimizer, training is done for 200 epochs. However, we also save the models the first time they achieve a training loss inferior or equal to 0.01, as done in [8], and during every epoch divisible by 50 (after the moment that loss of 0.01 is achieved).

For minimizing the objective during the calculation of minimum sharpness we use the SGD optimizer with a scheduler, as used in the original work of [6]. For the learning rate, we use 0.1 in the case of FashionMNIST and 0.5 in the case of CIFAR-10. The number of epochs is set to 100'000. In the case of divergence, we half the learning rate and double the number of epochs.

B. SAM contributes to generalization and converges to less sharp minima

In figure 1 we plot the test accuracies obtained after the epoch in which a loss of 0.01 is achieved. We observe that for every combination of dataset, architecture and background optimizer, SAM generalizes better than the background optimizer alone. The best results are mostly achieved by SAM with PHB as a background optimizer.



(a)

(b)

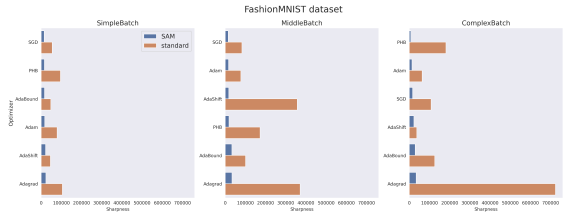
Figure 1: Test accuracies obtained for FashionMNIST 1a and CIFAR-10 1b, at the time when the loss reached a value of 0.01

In figure 2, we plot the values of sharpnesses of the model obtained after the epoch in which a loss 0.01 is achieved. We observe that for every combination of dataset, architecture and background optimizer, SAM converges to significantly less sharp minima than background optimizer alone. We comment that this is not obvious, because the notion of sharpness that SAM minimizes, is not the same as minimum sharpness. We believe this suggests that there exists a connection between these two different notions of sharpness, and can contribute to better understanding sharpness in general.

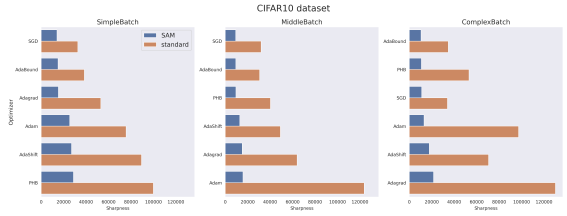
C. Correlation between minimum sharpness and generalization gap

The previous plots confirm that SAM generalizes better and converges to less sharp minima (based on minimum sharpness).

[2]https://github.com/ibayashi-hikaru/minimum-sharpness
[3]https://github.com/Luolc/AdaBound
[4]https://github.com/MichaelKonobeev/adashift



(a)



(b)

Figure 2: Test accuracies obtained for FashionMNIST 2a and CIFAR-10 2b, in the moment when training accuracy reached loss of 0.01.

This naturally leads to the question of the correlation between minimum sharpness and the generalization gap. Here we seek to give an answer to this question.

The generalization gap is calculated as the difference between the training and test accuracy. For a fixed dataset, we plot the sharpness as a function of the generalization gap and report the Kendall-Tau coefficient, as done in [8]. Results are show in the figure 3.

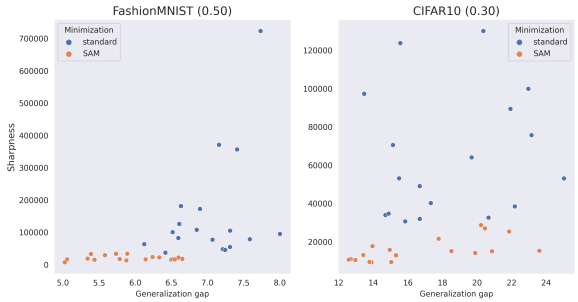


Figure 3: The connection between the generalization gap and the minimum shapness obtained on FashionMNIST and CIFAR-10. Minimum sharpness is computed when the loss has converged (i.e when it reaches a value of 0.01). In the brackets the Kendall-Tau coefficients are reported. Moreover, p-values of both coefficients were less than 0.05, so we these coefficients are significant.

Firstly, we again see that SAM finds converges to less sharp minima. Moreover, SAM and non-SAM trainings are almost linearly separable. Secondly, we see that the Kendall-Tau coefficient is larger in the case of FashionMNIST than in the case of CIFAR-10. We speculate that this happens because FashionMNIST is simpler than CIFAR-10. This speculation is also justified by the fact that on MNIST, that is the simplest of all the sets, the correlation seems to be higher [6]. Finally, we state that the Kendall-Tau coefficient between the generalization gap and various other different measures on CIFAR-10 can be found in [8]. For example, one of the best measures achieves around 0.5 correlation, while the path-norm measure yields a correlation coefficient similar to minimum sharpness.

D. Choice of optimizer - adaptive vs. non-adaptive

Finally, we tackle the question of the choice of optimizer, without considering SAM. In figure 4, we plot the distributions of test accuracies and sharpnesses per optimizer in the case of the CIFAR-10 dataset and sort them according to the mean. The same plot for FashionMNIST can be found in figure 8 in the appendix. We point out that, for both datasets, the order of the optimazers is the same: SGD, AdaBound, Adam, PHB, AdaShift, AdaGrad.

However, based on our experiments, it can not be deduced that adaptive algorithm generalize worse than non-adaptive ones, as stated in [1]. In order to check this, we performed the Kolmogorov-Smirnov test on the test accuracies obtained by non-adaptive and by adaptive algorithms. The p-value obtained is 0.48, so we can not conclude that the accuracies are drawn from different distributions. A similar result holds for the sharpness, in this case the p-value is 0.64. Thus, we can not say that adaptivity affects minimum sharpness.

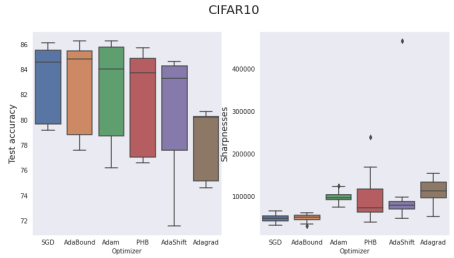


Figure 4: The distribution of accuracies and sharpnesses of the different optimizers used on CIFAR-10

IV. Conclusion

Using six different optimizers and SAM on the FashionMNIST and CIFAR-10 datasets, we obtained three main experimental results.

- Using the SAM optimization schemes yields a better generalization performance, as measured by the accuracy reached on the test data, compared to using the base optimizer alone.
- Minima reached by SAM are consistently less sharp, as measured by minimum sharpness, than the minima reached by the base optimizers alone. This holds, even though the notion of the sharpness minimized by SAM differs from that of minimum sharpness.
- There is a correlation of 0.5 on the FashionMNIST dataset, and 0.3 on the CIFAR-10 dataset between the generalization gap and the minimum sharpness, as measured by the Kendall-Tau coefficient. This correlation is not so strong, but still not neglectable.
- Our experiments give no conclusive evidence that adaptive algorithms generalize better than non-adaptive ones, neither that sharpness differs.

References

[1] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," *Advances in neural information processing systems*, vol. 30, 2017.

[2] R. M. Schmidt, F. Schneider, and P. Hennig, "Descending through a crowded valley-benchmarking deep learning optimizers," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9367–9376.

[3] 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.

[4] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1019–1028.

[5] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9636–9647.

[6] H. Ibayashi, T. Hamaguchi, and M. Imaizumi, "Minimum sharpness: Scale-invariant parameter-robustness of neural networks," *arXiv preprint arXiv:2106.12612*, 2021.

[7] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] 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.

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

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[11] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. null, p. 2121–2159, jul 2011.

[12] Z. Zhou, Q. Zhang, G. Lu, H. Wang, W. Zhang, and Y. Yu, "Adashift: Decorrelation and convergence of adaptive learning rate methods," *arXiv preprint arXiv:1810.00143*, 2018.

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

V. Appendix

A. Models (FashionMNIST)

Figure 5: Model architecture - SimpleBatch

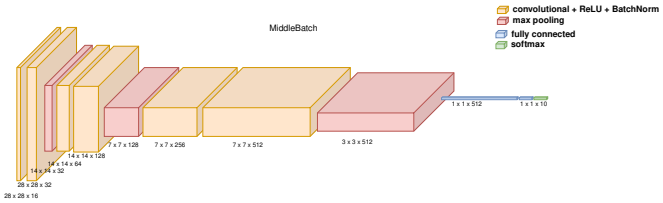


Figure 6: Model architecture - MiddleBatch

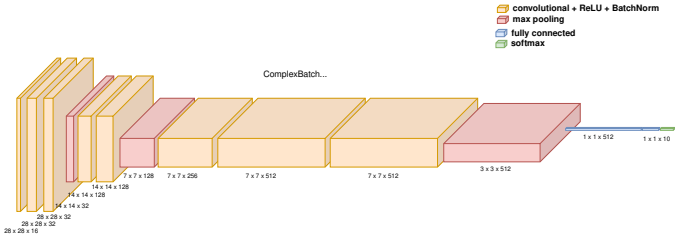


Figure 7: Model architecture - ComplexBatch

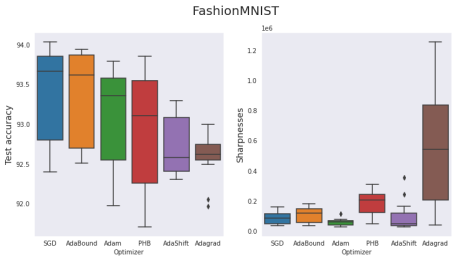


Figure 8: The distribution of accuracies and sharpnesses of the different optimizers used on FashionMNIST