Uppsala University

Numerical Optimization

FTN-0578

Optimization Project

Author:
Csongor Horváth

January 27, 2025

1 Introduction

The aim of this study to investigate the possible idea of Sharpness Aware Minimization (SAM) for free in the setting of deep learning. Original SAM algorithm was introduced by [2], motivated by earlier theoretical work on the dynamics of optimization in deep learning where it was observed that empirically in some sense more flat minima leads to better generalization performance [5, 6, 3]. It shall be noted, that for the simplest flatness measure, this correlation is only empirical and provably can't be causal due to the possible reparametrization of deep networks with a given activations such as ReLu [1].
A simply formulation for a flatness measure would be at parameter θ:

$$\max_{\|\epsilon\| \leq \rho} L(\theta + \epsilon) - L(\theta)$$

Here L is the empirical loss of a model. This sharpness was related to the problem formulation

$$min_{\theta^*} \, max_{\theta \in B_\rho(\theta^*)} \, L(\theta).$$

This is relevant as it can be shown that for any ρ with high probability for a training set S generated from distribution $\mathcal{D}$ this value is bounding the true expected loss in the following sense:

$$L_{\mathcal{D}}(\theta) \leq max_{\|\epsilon\| \leq \rho} \, L_S(\theta + \epsilon) + h(\|\theta\|/\rho^2).$$

In the original paper the loss derivative of the above minimization problem was approximated by first order approximation leading to an approximation of the loss:

$$\nabla_\theta \, L_S^{SAM}(\theta) := \nabla_\theta max_{\theta \in B_\rho(\theta^*)} \, L_S(\theta) \approx \nabla_\theta L_S(\theta + \hat{\epsilon}(\theta)),$$

where $\hat{\epsilon}(\theta) = \rho \, sign(\nabla_\theta L_S(\theta)) |\nabla_\theta L_S(\theta)| / (\|\nabla_\theta L_S(\theta)\|_2^2)$.

2 Research proposal

There has been many attempt to reduce the computational overhead created by the extra gradient computation of evaluating $\hat{\epsilon}(\theta)$, for details see [4]. In this project the possibility of fast approximation of the SAM loss is investigated. The idea is to approximate the gradient step in the t iterative step

$\nabla_\theta L_S^{SAM}(\theta)$ using the gradient of the last step $\nabla_\theta L_S(\theta_{t-1})$. As this gradient is necessary to compute and the memory overhead is not significant this could result in a computationless approximation of the SAM gradient.
Now let's motivate why we think it would be possible to make this approximation. For this let's explain the $\nabla_\theta L_S^{SAM}(\theta_t)$ computation. It looks like computing an inner step $\hat{\epsilon}(\theta_t)$, which means a locally upward (so gradient direction) step with a length of ρ. Now in the position after this inner step we compute the gradient and make a negative step in this gradient direction from θ_t.
Our argument is that if there is only slight changes in the gradient, so $\nabla_\theta L_S^{SAM}(\theta_{t-1}) \approx \nabla_\theta L_S(\theta_t)$, and the length of the $t.$ update was around ρ, this would mean that $\theta_t + \hat{\epsilon}(\theta_t) \approx \theta_{t-1}$. Therefore with continuous derivatives we have the approximation of the approximation $\nabla_\theta L_S^{SAM}(\theta_t) \approx \nabla_\theta L_S(\theta + \hat{\epsilon}(\theta)) \approx \nabla_\theta L_S(\theta_{t-1})$.
Let's make some note on the reality of our assumptions. During training the step length for the B batch from the training data S $\|\nabla_\theta L_B(\theta)\|$ was monitored and it is typically between 7 and 1 (for ResNet18 using the CIFAR10 dataset with batch of 512), so using standard learning rate for SGD, so between 5e-2 and 1e-3 we have a ρ value not too far of from the 0.05, which was choose as best in the original paper, but close ρ values within a magnitude had similarly good results. So we could argue that this is a way of dynamically changing the ρ value from step to step. It should be noted that changing ρ this way is counter intuitive as the guess would be we want larger ρ in flat areas and smaller ρ in steep parts and gradient length adaption does the reverse of this. This assumption seems feasible nevertheless.
The other assumption that $\nabla_\theta L_S^{SAM}(\theta_{t-1}) \approx \nabla_\theta L_S(\theta_t)$ might be more problematic. Here we argue that the most important part is when we are converging as we want to converge to flat parts, there this assumption should hold quite well. The problem could be only that in steep regions the direction of the convergence might differ significantly compered to SAM.

3 Feasibility study in a 2D test problem

To have a better feeling about the behavior of the proposed process we used a simple 2D test case on the flattened version of the Six-Hump Camel function. Based on these experiment it seems that the proposed algorithms (called FreeSAM) is even more likely to leave the region of a local minima, which

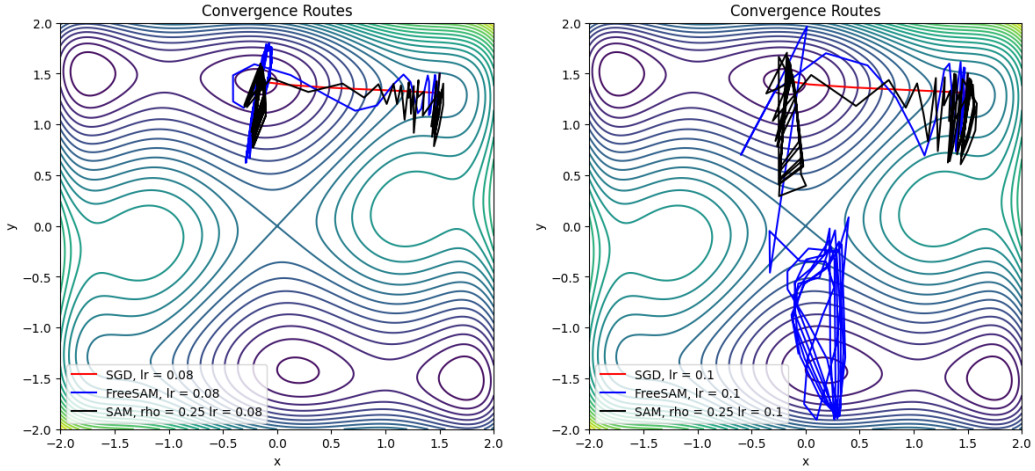

Figure 1: Convergence plot with smaller learning rate (left) and larger (right).

might make it even better for avoiding sharp minima.

Two example plot representing it with different learning rate can be found in Figure 1.

Note that these experiments not guarantee success in the setting of deep learning, but as the observed behavior is similar in these cases for SAM and the FreeSAM version with slight differences and both have the possibility of leaving local minima the proposal seems feasible to be tested on the setting of neural networks.

The code for this experiment can be found in `https://colab.research.google.com/drive/137OARFH2fWGxTZFyjfgC0KTliI47lnDi?usp=sharing`.

4 Demonstrating large scale feasibility for deep learning

The below presented results are biased in a sense that the original experiments with smaller networks and data sets doesn't show significant difference and benefit for using this method. Therefor for a more conclusive understanding many additional empirical and possibly theoretical studies are needed. Our aim here is just to demonstrate the feasibility and possible benefits produced by our method.

The final experimental setting was using the ResNet50 model for the

	tr loss	val loss	tr acc	val acc
SGD	$09e-4 \pm 2e-4$	2.09 ± 0.05	99.98 ± 0.002	69.92 ± 0.51
SAM	$38e-3 \pm 3e-3$	1.23 ± 0.02	99.58 ± 0.035	70.62 ± 0.11
FreeSAM	$16e-4 \pm 3e-4$	2.07 ± 0.07	99.96 ± 0.01	72.82 ± 0.49

Table 1: Average performance for different methods.

Cifar100 data set. There were only slight attempt made for optimize the learning rate and additional methods such as momentum or weight decay wasn't used in any of the cases, so the compared performance is not at the level of the state of the art. Furthermore our FreeSAM method seemed more sensitive to the chosen learning rate while SAM and SGD seemed more tolerant for it. The final choice of parameters based on some fine tuning was made to be 4e-2 for FreeSAM and 1e-2 for the other two method. For compute equivalent comparison the SAM method was only run for 100 epochs while the other two is for 200 epochs. Average results based on three initialization and random seeds are shown in table 1. Note that where there is larger fluctuation in the values such as some case in SGD and SAM, there not the final step, but one of the last step with highest test accuracy is shown (this is bit biased as there is no way for us to predict without the use of the training data which epoch is the best in performance).

Now to better understand the training process one-one instance of the training processes are compared step-by-step in terms of the four value in the figures 2

Based on these results we can argue the following, which was observed in other cases as well. Free SAM has a smooth convergence path in terms of epoch by epoch statistics similar or even better than the normal SAM method. Empirically in this test case this method has the fastest stable convergence in terms of the training data both for loss and accuracy.

For test loss it also converges the fastest, but doesn't share the good property of SAM to be able to keep down the test loss for a long period, but in this case we saw increased test loss more or less similar to the best cases in the SGD training run.

Despite SAM performing better in terms of loss, the FreeSAM method is able to achieve the best test accuracy with a significant margin.

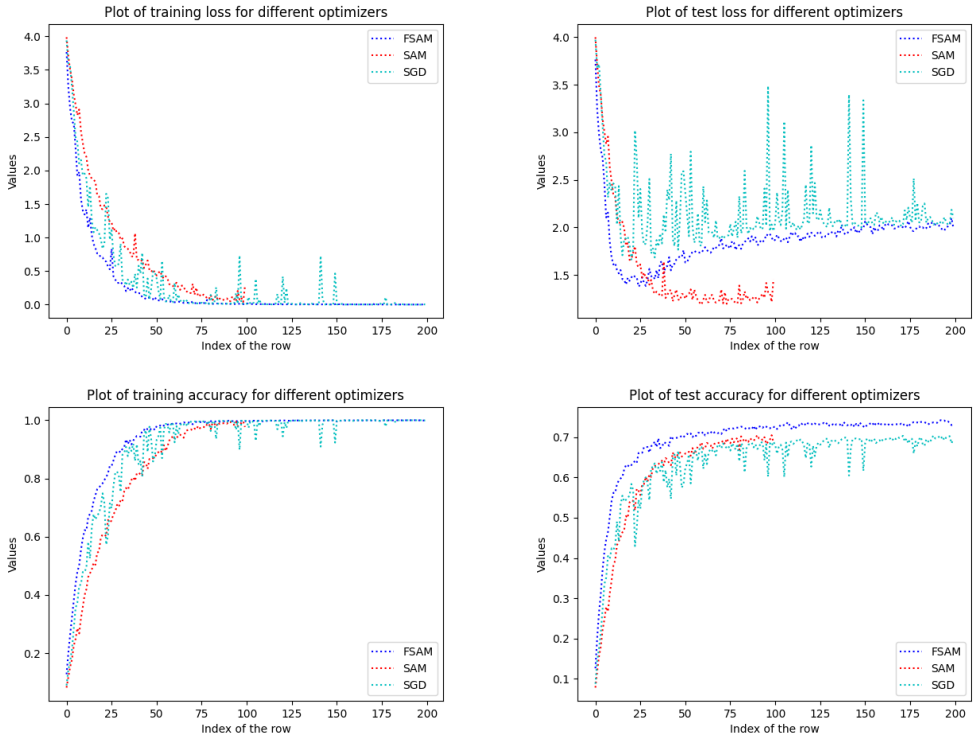

Figure 2: Dynamics plot for the optimization processes

5 Summary

We motivated a computationally free, and with memory usage only slightly more expensive approximation idea for the SAM method. We showcased the feasability in a 2D test case and in a deep learning setting.

Further work is necessary to decide under which conditions if any can our FreeSAM method outperform SGD and SAM similarly to the presented case. It is also worth further investigation to study the dynamics of optimization methods from the observed point of view that loss and accuracy can often be somewhat unrelated as in the shown case SAM is clearly the best in term of test loss and FreeSAM only slightly better than SGD. On the other hand considering the accuracy SAM only slightly better than SGD, while FreeSAM outperforms both with a significant margin once we choose a complex enough problem and suitable learning rate for the method.

Notes on code and contribution

The code reproducing these results and a script for an example run case is attached to the document submission.

The development of these results are based on a discussion with Li Ju and Aleksandr Karakulev. I appreciate their contributions to this work.

References

[1] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, *Sharp Minima Can Generalize For Deep Nets*.

[2] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, *Sharpness-Aware Minimization for Efficiently Improving Generalization*, Apr. 2021.

[3] S. Hochreiter and J. Schmidhuber, *SIMPLIFYING NEURAL NETS BY DISCOVERING FLAT MINIMA*, in Advances in Neural Information Processing Systems, vol. 7, MIT Press.

[4] C. Horváth, *Introduction to sharpness-aware minimization*, 2024.

[5] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, *Fantastic Generalization Measures and Where to Find Them*.

[6] 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*.