Sharpness-Aware Minimization (SAM)

Olga Gorbunova [1] **Farid Davletshin** [1]

Abstract

In our project, we decided to replicate the findings of the paper in which a method for dealing with the difficult task of training neural networks was proposed. This problem is associated with a loss function with a non-convex surface, which affects the solution's convergence to the locally optimal rather than the globally optimal. The authors described an algorithm that can account for this feature. We will put it to the test in this project.

Github repo: project link

1. Introduction

Modern models train through optimization methods, relying just on the training loss. These models can easily memorize the training data and are prone to overfitting (Chiyuan Zhang, 2016). They have more parameters than needed, and this large number of parameters provides no guarantee of proper generalization to the test set. According to recent empirical and theoretical studies, (Yiding Jiang, 2021), generalization is strongly related to the sharpness of the loss landscape at the learned parameter. Motivated by these studies, (Pierre Foret, 2021) proposes to penalize the sharpness of the landscape to improve the generalization.

Sharpness-Aware Minimization (SAM) is a procedure that aims to improve model generalization by simultaneously minimizing loss value and loss sharpness (the Figure 1 provide intuitive support for the notion of "sharpness" for a loss landscape).

In addition, in order to avoid catastrophic overfitting, there is a class of attack algorithms called momentum iterative gradient-based methods, in which gradients of the loss function are accumulated at each iteration to stabilize optimization and escape from poor local maxima (Yinpeng Dong, 2018). One of this class is fast gradient sign method (FGSM).

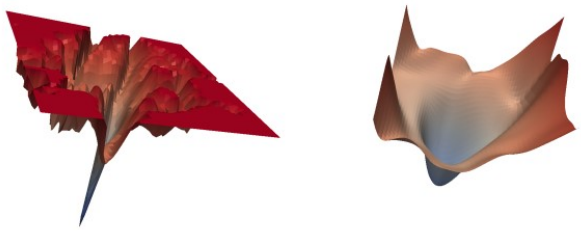

Figure 1. Sharp minimum (left) vs wide minimum (right)

The main contributions of this report are as follows:

- We efficiently implemented Sharpness-Aware Minimization (SAM)
- We trained ResNet-50 for a set of epochs on Cifar10 and Cifar100 datasets and compared the results with and without SAM
- Replaced optimization proposed in the (Pierre Foret, 2021) by finding an optimum ε via a mutiple step FGSM and insert it into the gradient evaluation for L_{SAM}
- We reviewed recent improvements and new ideas around SAM

2. Models and Algorithms

2.1. Models

The ResNet-50 model was used as the foundation for the experiments. This is a convolutional neural network with residual connections. Convolutional neural networks have generally performed well on image processing tasks, which is the task at hand. After comparing the amount of training data, a network with 50 layers was chosen from the Reset model family. We also tested larger models (ResNet-101 and

[1]Skolkovo Institute of Science and Technology, Moscow, Russia.

Final Projects of the Bayesian Methods of Machine Learning 2022 Course, Skoltech, Moscow, Russian Federation, 2020.

ResNet-152) but found that the ResNet-50 model provided the best balance of quality and training time.

2.2. Algorithms

According to the paper (Pierre Foret, 2021), loss can be represented by the sum of two terms. The first term captures the sharpness of L_S at w by measuring how quickly the training loss can be increased by moving from w to a nearby parameter value; this sharpness term is then added to the training loss value and a regularizer based on the magnitude of w.

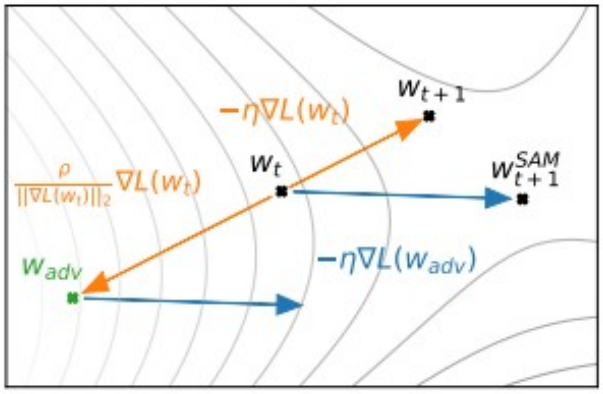

Figure 2. Schematic of the SAM parameter update.

The diagram above depicts how the SAM parameter update process differs from the norm.

From the paper, we got the following approximations:

$$\hat{\epsilon}(w) = \rho sign(\nabla_w L_{\mathcal{S}}(w))|\nabla_w L_{\mathcal{S}}(w)|^{q-1}/\left(\|\nabla_w L_{\mathcal{S}}(w)\|_q^q\right) \tag{1}$$

$$\nabla_w L_{\mathcal{S}}^{SAM}(w) \approx \nabla_w L_{\mathcal{S}}(w)|_{w+\hat{\epsilon}(w)} \tag{2}$$

We get the final SAM algorithm by using a standard numerical optimizer, such as stochastic optimization (SGD) to the SAM objective $L_S^{SAM}(w)$, with (2) used to compute the required objective function gradients.

To build the FGSM algorithm, we replaced the formula (2) for calculating $\hat{\epsilon}$ with the following:

$$\hat{\epsilon}(w) = \rho sign(\nabla_w L_{\mathcal{S}}(w)) \tag{3}$$

and did multi-step FGSM.

Algorithm 1 SAM algorithm

Input: Training set data $\cup\{x_i, y_i\}$, Loss function $l : W \times X \times Y \longrightarrow \mathcal{R}_+$, batch size b, step size η, neighborhood size ρ.
Output: Model trained with SAM.
Initialize weights $w_0, t = 0$.
while not converged **do**
  Sample batch $\mathcal{B} = \{(x_1, y_1), ...(x_b, y_b)\}$;
  Compute gradient $\nabla_w L_{\mathcal{B}}(w)$ of the batch's training loss;
  Compute $\hat{\epsilon}(w)$ per equation 1;
  Compute gradient approximation for the SAM objective (equation 2) $g = \nabla_w L_{\mathcal{B}}(w)|_{w+\hat{\epsilon}(w)}$;
  Update weights: $w_{t+1} = w_t - \eta g$;
  t = t + 1;
end while
return w_t

3. Experiments and Results

3.1. Experiments

Datasets We used the CIFAR-10 and CIFAR-100 datasets for our task, which are labeled subsets of the 80 million tiny images dataset.

The CIFAR-100 is made up of 100 classes, each with 600 images, 500 of which are training images and the rest are test images. We used the entire train and test sets as our train and test. The CIFAR-10 dataset is similar to the CIFAR-100 in that it contains 60000 32x32 color images divided into 10 classes, each with 6000 images. There are 50,000 test images and 10,000 training images. For the training set, we took 100 random samples with uniform distribution between classes, and for the test set, we took the entire test batch, which has 1000 images from each class chosen at random.

To reduce overfitting, we added numerous augmentations to the train set. They are as follows: random crop with four pixels padding, auto-augmentation, random horizontal flip, and cutout. All of these transformations were mentioned in the original article, and we decided to use them as well. Normalization was applied to CIFAR datasets for both train and test sets.

Main experiments All experiments have been conducted using Google Colab's and Datasphere computational resources. We used PyTorch as a framework for our project. If not stated otherwise, the optimizers were given the default hyperparameters.

We have trained 12 models with different numbers of epochs on CIFAR-10 and on CIFAR-100. The optimizer was set to SGD. We reproduced all hyperparameters from the original paper. Also, we present the general training pipeline in the

notebook.

3.2. Results

Tables (1) and (2) show the results of the testing of the SAM and without-SAM approaches (2). The SAM method significantly reduces error rates in the CIFAR-100 case and slightly improves quality in the CIFAR-10 case. For a large number of epochs, an amazing result has occurred: nothing is as predictable here. Without SAM, the algorithm produced a surprisingly large upgrade, whereas SAM performed worse. Below are graphs of learning curves for SAM and without-SAM algorithms. They demonstrate that in this particular case, SAM algorithm is much less stable.

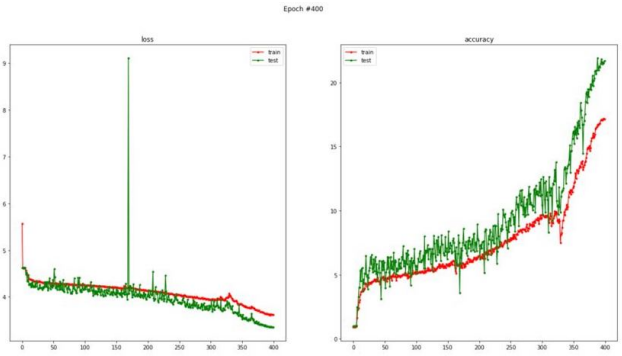

Figure 3. Training curves SAM on Cifar-100

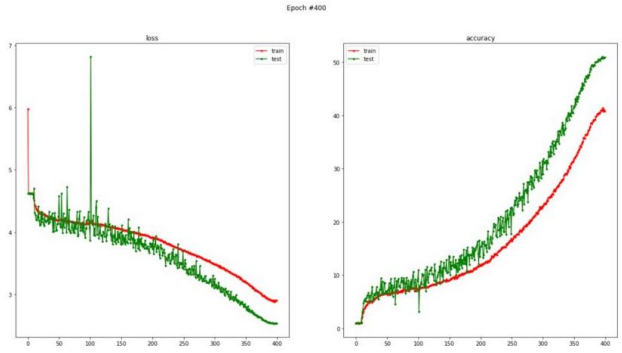

Figure 4. Training curves without-SAM on Cifar-100

We do not provide the results of the FGSM experiment because nothing useful could be obtained from it. The model was not trained, which is most likely due to implementation errors. However, we provide the algorithm in the repository; it works, but it falls short of expectations.

4. Literature review

4.1. ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks (Jungmin Kwon, 2021)

Authors state that sharpness defined in a rigid region with a fixed radius has a drawback in its sensitivity to parameter rescaling, which leaves the loss unaffected, but leading to a weakening of the connection between sharpness and the generalization gap. They suggest a novel learning method, adaptive sharpness-aware minimization (ASAM), utilizing the proposed generalization bound. Experimental results in various benchmark datasets show that ASAM contributes to a significant improvement in model generalization performance.

4.2. Generalized Federated Learning via Sharpness Aware Minimization (Zhe Qu, 2022)

Federated Learning (FL) is a promising framework for performing privacy-preserving, distributed learning with a set of clients. Many FL algorithms focus on mitigating the effects of data heterogeneity across clients by increasing the performance of the global model. The authors propose a general, effective algorithm, `FedSAM`, based on SAM local optimizer, and develop a momentum FL algorithm to bridge local and global models. The convergence analysis of these two algorithms was shown in the paper, and it was stated that the proposed algorithms significantly outperformed existing FL studies.

4.3. Make Sharpness-Aware Minimization Stronger: A Sparsified Perturbation Approach (Peng Mi, 2022)

SAM smooths the loss landscape by minimizing the maximized change in training loss when adding a perturbation to the weight. However, the authors believe that indiscriminate SAM perturbation on all parameters is suboptimal, resulting in excessive computation, i.e., twice the overhead of common optimizers like stochastic gradient descent. In this paper, the authors propose an efficient and effective training scheme coined Sparse SAM (SSAM), which achieves sparse perturbation by a binary mask. Sparse SAM not only has the potential for training acceleration but also smooths the loss landscape effectively.

5. Conclusion

On the CIFAR-10 and CIFAR-100 datasets, we showed how the SAM algorithm could be used. Due to some inconsistencies in the experimental results, we cannot state with certainty that this algorithm has outperformed the competition in all scenarios. During training, there might have been network issues, implementation errors, or just a human

Table 1. Test error rates for ResNet trained on CIFAR-10, with and without SAM.

CIFAR-10	EPOCH	TOP-1	TOP-K
NO SAM	100	15.34	0.91
NO SAM	200	12.94	0.89
NO SAM	400	11.24	**0.8**
SAM	100	**14.5**	**0.7**
SAM	200	**12.48**	**0.55**
SAM	400	**11.07**	1.08

Table 2. Test error rates for ResNet trained on CIFAR-100, with and without SAM.

CIFAR-100	EPOCH	TOP-1	TOP-K
NO SAM	100	89.93	66.33
NO SAM	200	90.19	66.19
NO SAM	400	**48.98**	**20.6**
SAM	100	**57.6**	**25.5**
SAM	200	**55.25**	**24.93**
SAM	400	78.09	43.27

factor. In these circumstances, it was necessary to repeat the experiment; however, due to the short time available (training 400 epochs takes about half a day), we make do with what we have.

At the same time, one should not underestimate how much better the SAM algorithm performs on the same CIFAR-100 but on epochs 100 and 200. This suggests that the algorithm actually contributes to higher learning quality.

Also, there are some open questions that can be considered as a continuation of this work:

- SAM is a new training method. Can we find some new architectures that are suitable for SAM?
- How can one make it more computationally effective? Right now, we have to compute the gradient twice in the loop (one approach is multi-step FGSM).
- Try to schedule ρ

References

Chiyuan Zhang, Samy Bengio, M. H. Understanding deep learning requires rethinking generalization, 2016. URL https://arxiv.org/abs/1611.03530.

Jungmin Kwon, Jeongseop Kim, H. P. Sharpness-aware minimization for efficiently improving generalization. *ICLR Spotlight*, 2021. doi: 10.1186/s40537-021-00444-8. URL http://proceedings.mlr.press/v139/kwon21b.html.

Peng Mi, Li Shen, T. R. Y. Z. Make sharpness-aware minimization stronger: A sparsified perturbation approach, 2022. URL https://openreview.net/forum?id=88_wNI6ZBDZ.

Pierre Foret, Ariel Kleiner, H. M. Sharpness-aware minimization for efficiently improving generalization. *ICLR Spotlight*, 3(1), March 2021. doi: 10.1186/s40537-021-00444-8. URL https://arxiv.org/abs/2010.01412.

Yiding Jiang, Behnam Neyshabur, H. M. Fantastic generalization measures and where to find them, 2021. URL https://arxiv.org/abs/1912.02178.

Yinpeng Dong, Fangzhou Liao, T. P. Boosting adversarial attacks with momentum, 2018. URL https://arxiv.org/pdf/1710.06081.pdf.

Zhe Qu, Xingyu Li, R. D. Y. L. Generalized federated learning via sharpness aware minimization. *ICLR Spotlight*, 2022. doi: 10.1186/s40537-021-00444-8. URL https://proceedings.mlr.press/v162/qu22a.html.