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).

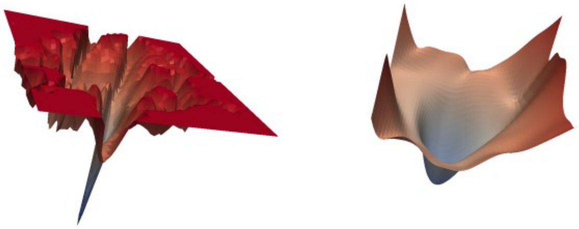

Figure 1. Sharp minimum (left) vs wide minimum (right)

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.

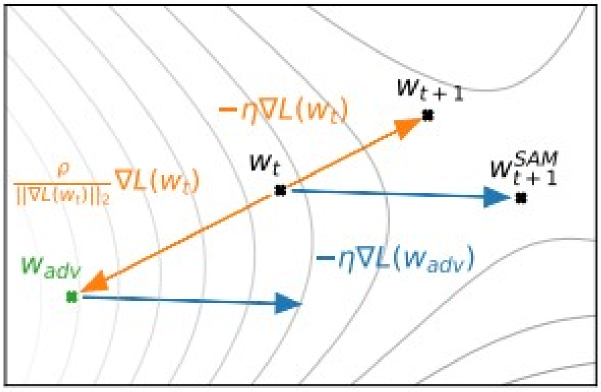

Figure 2. Schematic of the SAM parameter update.

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

From the paper, we got the following approximations:

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

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

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:

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

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 $\eta$, neighborhood size $\rho$.
  **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 For our task, we used the CIFAR-10 and CIFAR-100 datasets, which are labeled subsets of the 80 million tiny images dataset.

The CIFAR-100 consists of 100 classes containing 600 images each, from which 500 images are train ones and the rest are test images. We took the whole train and test sets as our train/test, respectively. Our data preprocessing of CIFAR-100 involved only normalization with mean 0.5 and std 0.5 per each channel.

The CIFAR-10 dataset is similar to the CIFAR-100, but it contains 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. We took 100 random samples for the train set with uniform distribution between classes and for our test set we took the whole test batch, which has 1000 images from each class, selected randomly. For augmentation of the train set we used composition of random crops and horizontal flips and finally normalized the images with the same mean and s.t.d. as for CIFAR-100, but for the test set we only applied normalization.

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

First, we have trained 12 models with different number of epochs on CIFAR-10 and on CIFAR-100. The optimizer was set as SGD

3.2. Results

We can see that

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 sensitivity to parameter re-scaling which leaves the loss unaffected, leading to weakening of the connection between sharpness and 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 significant improvement of 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. Authors propose propose a general, effective algorithm, `FedSAM`, based on SAM local optimizer, and develop a momentum FL algorithm to bridge local and global models. In the paper was shown the convergence analysis of these two algorithms and stated that proposed algorithms substantially outperform existing FL studies.

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

SAM smooths the loss landscape via minimizing the maximized change of training loss when adding a perturbation to the weight. However, authors find the indiscriminate perturbation of SAM on all parameters is suboptimal, which also results in excessive computation, i.e., double the overhead of common optimizers like Stochastic Gradient Descent. In this paper, we propose an efficient and effective training scheme coined as 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

We have succeeded in demonstrating the

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

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.