Sharpness-Aware Minimization for Efficiently Improving Generalization: A Review

Kaizhao Liu **Ruitao Chen** **Zhekun Shi**

Abstract

This a review of Sharpness-Aware Minimization (SAM) for Efficiently Improving Generalization[1]. We describe the background knowledge that induce SAM, compare it with previous results, and hence show the novelty and importance of SAM. Then we summarize and criticise the main results, develop some of them in our own means, and present two small-scale experimental examples. Finally we summarize the limitation of this paper and indicate future directions. Some of which have already been studied so we also present reference to them.

1 Background

Many modern machine learning models operate in overparameterized regime, especially deep learning models. As a consequence, there are many solutions of the corresponding optimization problem, and their congregations are much different from those in underparameterized regime . Usually, the global minima form a manifold [2]. Although classical optimization algorithms can converge to global minima, their generalization ability varies. We always want to find a minimum that generalize better.

Scholars had found that the landscape of loss function is related to generalization. Explicitly, *flatter* minima generalize better [3]. However, practical algorithms that directly seek out flatter minima had been elusive. Sharpness-aware minimization fills out this blank.

Researchers have long been developing algorithms to generalize better. For example, Entropy-SGD[4] minimizes local entropy, a concept which is related to sharpness. However, it is rather computationally expensive, involving calculation of integration. Moreover, the concept of local entropy is not very popular and its relationship with sharpness is complicated. Another approach is by averaging multiple points along the trajectory of SGD with cyclical learning rates[5].This approach adjusts each step by averaging with a decreasing learning rate in a cycle, which is related to sharpness implicitly by exploring the geometry of loss landscape using information from previous steps. But this implicit connection is hard to quantify. There is also an optimization method by diffusion[6]. This method creates a series of diffused problem, optimizes the ultimate diffused problem which is convex, and then uses this solution to initialize the less diffused problem to trace back. This is actually a continuation method, which is rather complicated, and its relationship with generalization remains mysterious. So compared with these results, sharpness-aware minimization is simple both computationally and conceptually. It concerns sharpness directly, thus improving generalization ability.

Here is the notation we use throughout our review. Suppose $l : \Theta \times \mathcal{X} \times \mathcal{Y} \to [0, 1]$ is the per-data loss, where $\Theta \subset \mathbb{R}^d$ denotes the parameter space. If we have a population $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, then the corresponding population risk is $L_D(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}} l(\theta, x, y)$. If we have a sample $S := \bigcup_{i=1}^n \{(x_i, y_i)\}$, then the corresponding empirical risk is $L_S(\theta) := \frac{1}{n}\sum_{i=1}^n l(\theta, x_i, y_i)$.

Preprint. Under review.

2 Main Results

In the original paper, the authors proposed a generalization bound and an algorithm. Then they demonstrated some empirical study of the algorithm. The core result is the algorithm, namely SAM (Equation 3).

2.1 Algorithm

In the original paper, following the paradigm of structural risk minimization, the authors were motivated by a PAC-Bayesian bound, and did several approximation to obtain a computationally-efficient algorithm which minimizes that bound.

2.1.1 Approaching SAM

We think the original approach is rather nonintuitive. Our approach is based on the intuitive observation that *flatter* minima generalize better, and then arrives at the same algorithm by penalizing flatness. First we need to quantify *flatness*.

Definition 1 (sharpness)**.** *Let $f : \Theta \to \mathbb{R}_+$ be a function and $\rho \geq 0$ be a hyperparameter. The sharpness of f at θ with hyperparameter ρ is the following quantity:*

$$\kappa_\rho(\theta) := \max_{\|\epsilon\| \leq \rho} f(\theta + \epsilon) - f(\theta) \tag{1}$$

where $\|\cdot\|$ is the standard Euclidean distance.

Informally, the sharpness in our context is defined to be the maximal variation if we sit at θ and measure within a ball with radius ρ. Note that the original paper used L^p norm, but we think that this generalization is unnecessary. So we keep focus on L^2 norm.

Example 1. *Suppose f is second order differentiable and $\nabla_\theta f \neq 0$. If ρ is small, we can approximate $f(\theta + \epsilon)$ by $f(\theta) + \epsilon^T \nabla_\theta f$. Maximizing gives $\hat{\epsilon} = \rho \frac{\nabla_\theta f}{\|\nabla_\theta f\|}$, so $\kappa_\rho(\theta) \approx \rho \|\nabla f(\theta)\|$.*

Example 2. *Suppose f is second order differentiable and θ is a global minimum. Then $\nabla_\theta f = 0$. If ρ is small, we can approximate $f(\theta + \epsilon)$ by $f(\theta) + \epsilon^T (\nabla^2_\theta f)\epsilon$. Suppose the maximal eigenvalue of Hessian $\nabla^2_\theta f$ is λ_1, and the corresponding eigenvector is v_1. Then maximizing gives $\hat{\epsilon} = \rho v_1$ and $\kappa_\rho(\theta) \approx \rho^2 \lambda_1$.*

From the above Example 2, it can be seen that our definition of sharpness roughly coincide with the maximal eigenvalue of Hessian, which is another popular definition of sharpness.

Now we formulate sharpness-aware minimization problem. We add a sharpness penalization term to the original empirical risk minimization, and yield:

$$\min_\theta \quad L_S(\theta) + \kappa_\rho(\theta) \Longleftrightarrow \min_\theta \quad \max_{\|\epsilon\| \leq \rho} L_S(\theta + \epsilon) \tag{2}$$

Note that our formulation omit the L^2 penalization term.

To derive the SAM algorithm, we just use gradient descent to solve the above problem. Now we need to compute $\nabla_\theta \max_{\|\epsilon\| \leq \rho} L_S(\theta + \epsilon)$. By the approximate sharpness $\kappa_\rho(\theta)$ in Example 1, when we have not reached global minima, $\nabla L_S(\theta) + \rho \nabla \|\nabla L_S(\theta)\|$. Although this expression reveals explicitly it is computationally inefficient. Instead, we use $\hat{\epsilon} = \rho \frac{\nabla_\theta L_S}{\|\nabla_\theta L_S\|}$ in Example 1, so we get $\nabla_\theta L_S(\theta + \hat{\epsilon}) = \frac{\mathrm{d}(\theta + \hat{\epsilon})}{\mathrm{d}\theta} \nabla L_S|_{\theta + \hat{\epsilon}} \approx \nabla L_S|_{\theta + \hat{\epsilon}}$ up to second order terms. This expression is much easier to compute, which only requires gradient information. So the final SAM update looks like:

$$x_{t+1} = x_t - \eta \nabla L_S|_{\theta + \rho \frac{\nabla_\theta L_S}{\|\nabla_\theta L_S\|}} \tag{3}$$

Here we find the update equation above invalid if $\nabla_\theta L_S = 0$, for this term appears in the denominator. Furthermore, the update stops when $\nabla L_S|_{\theta + \hat{\epsilon}} = 0$, which is not equivalent to $\nabla L_S|_\theta = 0$. These observations raise questions about the convergence of SAM. Furthermore, the error of the approximation above can not be controlled. In fact, the problems differ before and after the approximation, and we need to inspect them separately.

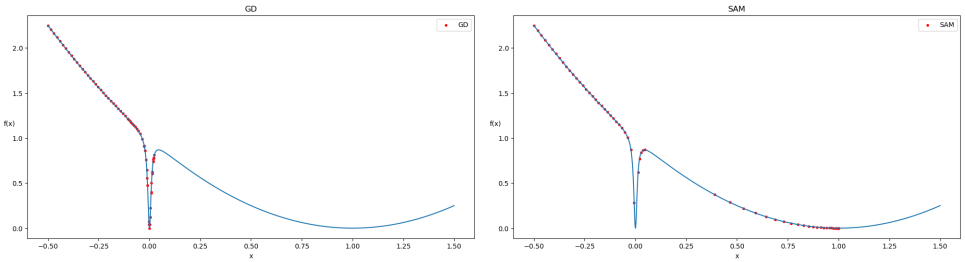

Figure 1: We use both SAM and simple GD to find the minima of function $\frac{x^2(x-1)^2}{x^2+0.0001}$ whose global minima are $x = 1$ (flatter) and $x = 0$ (sharper). When SAM and GD are both converged, it is clearly that SAM is hard to stop when the landscape is sharp. SAM can go through the sharp landscape and get a flatter minima. But simple GD sometimes may stop when the landscape is sharp so that it can not get a flatter minima.

2.1.2 Understanding SAM

Now we provide an understanding of why SAM can find flatter minimum. We derive an approximation to $\nabla L_S|_{\theta+\hat{\epsilon}}$. Compare it to gradient descent (GD):

$$x_{t+1} = x_t - \eta \nabla L_S|_\theta \tag{4}$$

We want to find the discrepancy between SAM and GD. Assume that the Hessian $\nabla^2 L_S(\theta)$ has decomposition $\nabla^2 L_S(\theta) = P^{-1}DP$ where D is the diagonal matrix with eigenvalues of the Hessian $\nabla^2 L_S(\theta)$. By Taylor expansion, we have

$$\begin{aligned} \nabla L_S|_{\theta+\hat{\epsilon}} - \nabla L_S|_\theta &\approx \nabla^2 L_S(\theta)\hat{\epsilon} \\ &= \nabla^2 L_S(\theta) \frac{\rho}{\|\nabla_\theta L_S\|} \nabla_\theta L_S \\ &= P^{-1}(\frac{\rho}{\|\nabla_\theta L_S\|} D) P \nabla_\theta L_S \end{aligned} \tag{5}$$

Thus, we can find that intuitively the discrepancy increases when ρ increases. SAM is exactly GD when $\rho = 0$. When $\|\nabla_\theta L_S\|$ is large, SAM is similar to GD which makes the loss L_S decrease. If $D = \lambda I$ which means all the eigenvalues of the Hessian $\nabla^2 L_S(\theta)$ are the same, then the direction of $P^{-1}(\frac{\rho}{\|\nabla_\theta L_S\|} D) P \nabla_\theta L_S$ is same as $\nabla_\theta L_S$. The magnitude of $P^{-1}(\frac{\rho}{\|\nabla_\theta L_S\|} D) P \nabla_\theta L_S$ increases when $|D|$ increases. Hence, when the discrepancy between eigenvalues of the Hessian $\nabla^2 L_S(\theta)$ is large or these eigenvalues are large, which means the landscape is shaped to some extent, SAM is not similar to GD. Thus, SAM is hard to stop when the landscape is sharp and is more likely to stop when the landscape is flat (Figure 1). Features of eigenvalues of the Hessian $\nabla^2 L_S(\theta)$ are very important and SAM can reflect these features to some extent.

2.2 Generalization Bound

The original paper provided a generalization bound:

Theorem 1. *Assume $L_{\mathcal{D}}(\theta) \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\rho I_d)} L_{\mathcal{D}}(\theta + \epsilon)$. Let d be the number of parameters. For any $\rho > 0$ and any distribution $\mathcal{D}$, with probability $1 - \delta$ over the choice of training set $S \sim \mathcal{D}$,*

$$L_{\mathcal{D}}(\theta) \leq \max_{\|\epsilon\| \leq \rho} L_S(\theta + \epsilon) + \sqrt{\frac{d \log(1 + \frac{\|\theta\|^2}{\rho^2}(1 + \sqrt{\frac{\log n}{d}})^2) + 4 \log \frac{n}{\delta} + 8 \log(6n + 3d)}{n - 1}}$$

The author obtained this bound on the basis of PAC-Bayesian bound.

Lemma 1 (PAC-Bayesian bound)**.** *Given a prior distribution $\mathcal{P}$ over Θ and for any posterior distribution $\mathcal{Q}$ over Θ, with probability $1 - \delta$ over the choice of sample S,*

$$\mathbb{E}_{\theta \sim \mathcal{Q}}(L_D(\theta) - L_S(\theta)) \leq \sqrt{\frac{KL(\mathcal{Q}\|\mathcal{P}) + \log \frac{n}{\delta}}{2(n-1)}}$$

The assumption $L_{\mathcal{D}}(\theta) \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\rho I_d)} L_{\mathcal{D}}(\theta + \epsilon)$ is satisfied if $L_{\mathcal{D}}(\theta)$ is convex and θ is a minimum. But for a general minimum, the exact meaning of this condition needs to be further discussed. The expectation captures sharpness in an averaging way, which is different from our definition 1 of sharpness.

In the original paper the authors derived SAM based on this generalization bound, but we do not think it convincing. The derivation of this bound is rather artificial. It involves Gaussian prior and posterior, and an artificial predefined set of prior variance. We think it can be vacuous and can only be viewed as a indication of 'flatter minima generalize better'. However, many PAC-Bayesian type bounds can serve this. So this specific bound in the original paper is not that meaningful. Usually we develop generalization bound for a certain hypothesis class to guarantee that we can avoid the curse of dimension. But this bound gives us no such guarantee for overparameterized regime where $d > n$. Nevertheless, it still provides some insight on the relationship between sharpness and generalization compared with other articles introducing new algorithms, which we have mentioned in Section 1.

2.3 Experiments

In the original paper, the authors did three sets of experiments. The first was to implement SAM on common dataset and showed that SAM improves generalization performance in most of the cases. The second set was to apply SAM to finetuning on a smaller target data set of interest (pretrained by a model on a large related data set).The authors used EfficientNet-b7 (pretrained on ImageNet) and EfficientNet-12 (pretrained on ImageNet plus unlabeled JFT; input resolution 475) to show the effect of SAM in finetuning. The experiments showed SAM uniformly improves performance relative to finetuning without SAM. The third set studied the influence of label noise on SAM and found that SAM also improves robustness.

When utilizing SAM in practice, the authors gave a concept called 'm-sharpness', which is just sharpness measured with batch size m rather than the total train set size n. In order to scale training for many of today's models (always have big training set), we need to compute SAM faster. 'm-sharpness' gives us a good method to scale SAM because we use a part of training set instead of the whole training set, just like SGD over GD. Surprisingly, the authors found 'm-sharpness' also connects with the generalization. As m decreases, we may get a minima which has better generalization ability. In our opinion, we think that SAM with 'm-sharpness' can be regarded as a stochastic version of SAM. SAM with 'm-sharpness' uses a subset of the whole train set to finish updating per-batch, so SAM with 'm-sharpness' depends less on the training set. Instead, it samples from the empirical distribution, which is an approximation of the true data distribution. As batch size m decreases, we may depend less on the full training set. In a way, this is similar to the principle of bootstrap. Moreover, the less m is, the more random one update is, so the algorithm oscillates more around the minimum. If the minimum is sharp, it is easy for SAM with 'm-sharpness' to escape. So SAM with 'm-sharpness' prefers flatter minima.

2.3.1 Example: Separation between SAM and GD

First, we do a simple experiment which shows the separation between GD and SAM. We use GD and SAM to find the minima of function $f(x, y) = (x - 1)^2 y^2$, whose global minima are on the line $x = 1$ and $y = 0$. In both SAM and GD, we let learning rate $\eta = 0.1$. In SAM, we let the hyperparameter ρ=0.00001. We run both SAM and GD for 1000 iterations in order that both SAM and GD converge. From the experiment results, we find that SAM truly prefers flatter minima (more details in Figure 2). This shows empirically that relative to GD, SAM truly prefer flatter minima.

2.3.2 Performance on MNIST Dataset

We compare SAM to stochastic gradient descent (SGD)[7] and ADAM[8]. SGD is similar to SAM. Specifically, it is a special case of SAM with $\rho = 0$. Note that the author didn't compare SAM to ADAM in their original paper.

We apply them to train a model aimed to classify MNIST dataset. The model is a three-layer fully-connected network with widths $784 \to 500 \to 300 \to 10$. The loss function is softmax and crossentropy. We use 1000 samples as training set and evaluate the model on testing set with 10000

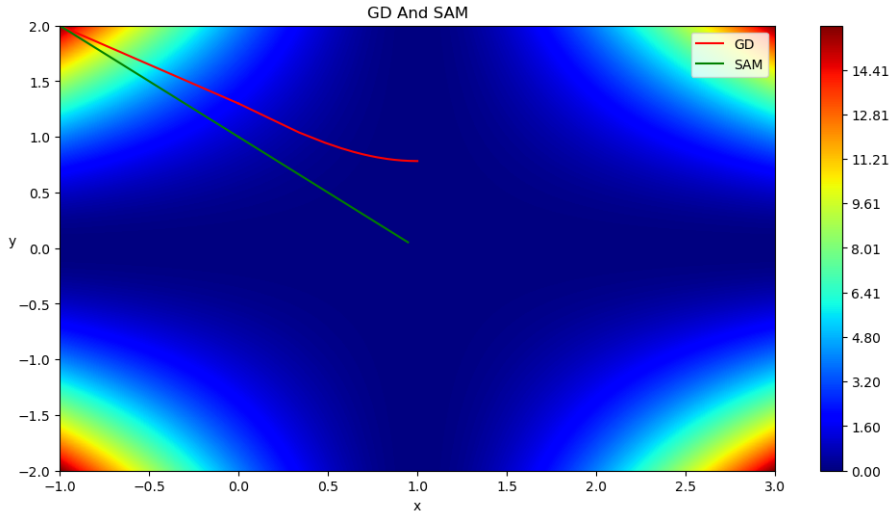

Figure 2: We can find that SAM truly finds flatter minima : SAM and GD both give a solution which is close to a certain minima but the solution given by SAM is closer to $(1, 0)$, which is a flatter minima than the solution of GD (the solution given by SAM has smaller sharpness(see Definition 1) than the other solution). We can calculate the max eigenvalue of $\nabla^2 f(x_{GD}) = 1.225$ and the max eigenvalue of $\nabla^2 f(x_{SAM}) = 0.016$. λ_{max} of $\nabla^2 f(x_{GD})$ is much greater than λ_{max} of $\nabla^2 f(x_{SAM})$, which can strongly prove that in this experiment SAM can truly find flatter minima

samples. For all algorithms, we set batch size to 10.We run all three algorithms for 15000 times of gradient computations. Thus, the model are trained by SGD or ADAM for 150 epochs. Note that we need to compute the gradient twice in one iteration of SAM. Thus, We only run SAM for 75 epochs.

We report the best performances for each algorithms in 5 independent runs.

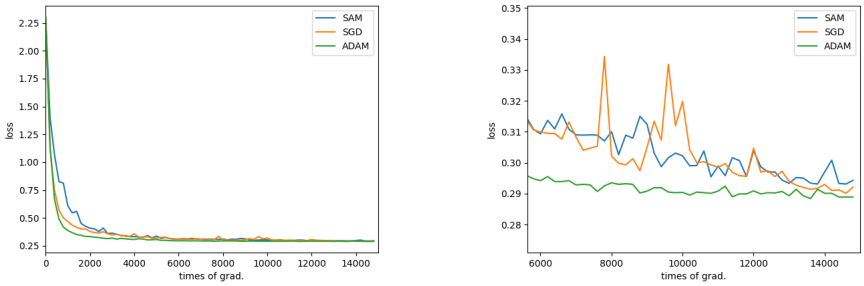

Figure 3: Evolution of loss in training of three algorithms

As seen in (Figure 3). SAM and SGD use similar epochs to converge which means that they have similar convergence rate according to epochs. However, the convergence rate according to CPU time of SAM is slower than SGD mainly because SAM computes the gradient twice in one iteration. ADAM converges much faster than these two algorithms. The training loss of SAM is similar to that of SGD. But ADAM reaches a better training loss.

The performances (testing loss) of three algorithms are shown in (Figure 4). SAM significantly outperforms SGD. However, SAM has no advantages compared to ADAM.

As seen in (Figure 5). Compared to SGD, SAM improves the generalization a lot. The generalization gap of SAM is only about 80% of that of SGD. But ADAM still outperforms SAM. Generalization of traditional algorithms like ADAM may be better than that of SAM in some cases.

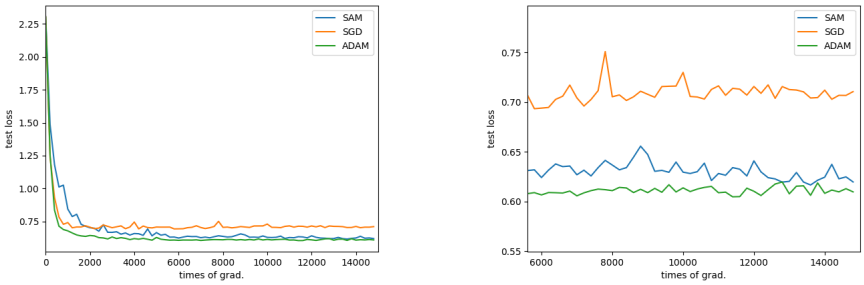

Figure 4: Evolution of loss in testing of three algorithms

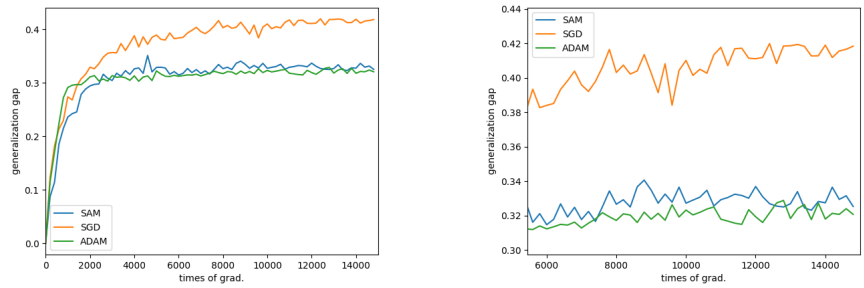

Figure 5: Evolution of generalization gap of three algorithms

3 Potential Research Directions and Some Realization

3.1 Before and After Approximation

Recall that when we derive SAM, we do several approximations to ease computational effort. But before and after the approximations, problems are essentially different. The solution of SAM problem is not equivalent to the solution of SAM algorithm. Moreover, they are all different from the problem addressed in the PAC-Bayesian generalization bound, which considers minimizing an expectation form of sharpness. We need a detailed analysis of this subtle difference in the future.

3.2 Convergence Property of SAM

Naturally, future works need to analyse the convergence and convergence rate of SAM. We find several authors have already analyze the convergence of SAM under some assumptions and give the convergence rate $O(\log T/\sqrt{T})$ (see the following theorem from [9]):

Theorem 2. *Suppose the stochastic SAM can be written as :* $\mathbf{x_{t+1}} = \mathbf{x_t} - \eta \cdot g(\mathbf{x_t} + \rho \cdot \frac{g(\mathbf{x_t})}{\|g(\mathbf{x_t})\|})$, *here* $g(\mathbf{x})$ *is an unbiased estimate of* $\nabla f(\mathbf{x})$.
Assumption 1. *(Bounded Gradient) It exists* $G \geq 0$ *s.t.* $\|\nabla f(\mathbf{x})\| \leq G$.
Assumption 2. *(Bounded Variance) It exists* $\sigma \geq 0$ *s.t.* $\mathbb{E}[\|g(\mathbf{x}) - \nabla f(\mathbf{x})\|^2] \leq \sigma^2$.
Assumption 3. *(L-smoothness) It exists* $L \geq 0$ *s.t.* $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$
Consider function $f(\mathbf{x})$ *satisfying above assumptions optimized by SAM, Let* $\eta_t = \frac{\eta_0}{\sqrt{t}}$ *and hyperparameter* ρ *decay with square root of* t, *e.g.* $\rho_t = \frac{\rho_0}{\sqrt{t}}$. *With* $\rho_0 \leq G\eta_0$, *we have*

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla f(\mathbf{x_t})\|^2 \leq C_1\frac{1}{\sqrt{T}} + C_2\frac{\log T}{\sqrt{T}}, \tag{6}$$

where $C_1 = \frac{2}{\eta_0}(f(\mathbf{x_0}) - \mathbb{E}f(\mathbf{x_T}))$ *and* $C_2 = 2(L\sigma^2\eta_0 + LG\rho_0)$.

Also, we think that the hyperparameter ρ plays an important role in SAM. This is a hyperparameter that connects global properties and local properties. Different values of ρ may cause different

convergence rate in SAM. In our opinion we think that ρ can be regarded as tradeoff between minimization of loss and minimization of flatness to some extent. Large ρ obtains a flatter minimum but with greater loss and small ρ obtains a sharper minimum but with lower loss (when $\rho \longrightarrow 0$, we get GD or SGD which don't consider flatness). In the future, we can analyze more about how hyperparameter ρ influences SAM.

3.3 Bias of SAM

What kind of bias does SAM possess? In what sense can we guarantee SAM converge to a flatter point? We need more accurate account of these implicit regularization effect of SAM. For example, there are many different version of sharpness. Which of them characterize SAM's bias? And we can also analyze the bias of SAM before the approximation.

3.4 Randomness and Generalization

The impact of m-sharpness on generalization need further inspection. m-sharpness is a stochastic version of common-used sharpness. It encodes sharpness in randomly oscillating around the original minimum, and its expectation can be regarded as the expectation formulation of sharpness mentioned in the PAC-Bayesian generalization bound. We can consider its variance and infer that we can generalize better if the variance is smaller.

3.5 Sharpness and Generalization

Connection between geometry and generalization needs further inspection. How are sharpness connected to generalization? We have already seen that PAC-Bayesian bound alone can not yield a satisfactory explanation. Maybe information about the specific model is needed.

3.6 Sharpness and Robustness

Recall that in the original paper the author claimed that SAM can improve robustness, or the stability under label pollution. How is it possible? Is sharpness of loss landscape preserved after label pollution? We need a mathematical framework to explain this phenomenon.

3.7 Different Generalization Performance of Different Algorithms

There are many articles focusing on empirical or theoretical results about generalization of tradition algorithms. For example, SGD may obtain better generalization than ADAM[10]. In our experiment, why ADAM performs better than SAM is not clear. Empirical and theoretical comparisons between new algorithms aimed to improve generalization and traditional algorithms are still important. And we want to know: using this sort of knowledge, can we develop a better algorithm to improve generalization? In addition, how can we choose the suitable training algorithm which has the best ability of generalization when facing different data sets and different models?

3.8 Better algorithm based on SAM

Based on the original paper, several papers give some variants of SAM. For example, there exists an efficient and effective training scheme coined as Sparse SAM(SSAM)([9]), which achieves sparse perturbation by a binary mask. And due to different methods of obtaining the sparse training mask, there exists two training scheme called SSAM-F and SSAM-D([9]). Furthermore, can we imagine better algorithm based on the notion of sharpness?

References

[1] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *International Conference on Learning Representations 2021*, 2021.

[2] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1309–1318. PMLR, 10–15 Jul 2018.

[3] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv e-prints*, 2016.

[4] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: biasing gradient descent into wide valleys*. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, dec 2019.

[5] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[6] Hossein Mobahi. Training recurrent neural networks by diffusion. *arXiv preprint arXiv:1601.04114*, 2016.

[7] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[9] Mi P, Shen L, Ren T, and et al. Make sharpness-aware minimization stronger: A sparsified perturbation approach[j]. *arXiv preprint arXiv:2210.05177, 2022.*

[10] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Hoi, and E. Weinan. Towards theoretically understanding why sgd generalizes better than adam in deep learning. 2020.