BN-invariant sharpness regularizes the training model to better generalization

immediate

Abstract

It is arguably believed that flatter minima can generalize better. However, it has been pointed out that the usual definitions of sharpness, which consider either the maxima or the integral of loss over a δ ball of parameters around minima, cannot give consistent measurement for scale invariant neural networks, e.g., networks with batch normalization layer. In this paper, we first propose a measure of sharpness, BN-sharpness, which gives consistent value for equivalent networks under BN. BN-sharpness achieves the property of scale invariance by scaling the integral diameter according to the magnitude of parameter. We then present a computation-efficient way to calculate the BN-sharpness approximately i.e., one dimensional integral along the "sharpest" direction. Furthermore, we use the BN-sharpness to regularize the training and design an algorithm to minimize the new regularized objective. Our algorithm achieves considerably better performance than vanilla SGD over various experiment settings.

1 Introduction

With the support of big data, deep learning has achieved a huge success across domains. In recent years, it becomes a common sense that deep neural networks have perfect training performance. Specifically, [25] empirically shows that the popular neural networks (NN) can almost always reach zero training error at the end of training process. Moreover, [7] and [2] prove that, for the neural networks with sufficient width, gradient descent converges to the global minima of the objective, under some conditions on the data generation. If training deep neural network is not a problem, how to find minima that generalize well becomes critical.

It's arguably believed that minimizer locates in a flat valley generalizes better. [20] proves a generalization bound given by "expected sharpness" based on the PAC Bayes theory in [16]. This viewpoint is also empirically confirmed by [12] and [15]. However, for neural network with scale invariance, the definitions of sharpness become problematic.

One important case is deep neural network with *batch normalization* (BN) [10]. BN that normalizes the input to a neuron in the network by its mean and standard deviation over a mini-batch of training data, is a widely used trick in training deep neural network. Batch normalization brings scale invariance to neural networks, i.e., scaling the parameter tying to one BN operation does not affect the output of the network. This scale invariant property of neural networks with BN makes the definition of sharpness problematic, i.e., two equivalent networks under BN have different sharpness, and creates trouble to the relation between sharpness and generalization. Then can we develop a scale-invariant sharpness for neural network with batch normalization? Meanwhile, can we leverage such sharpness to help us find minima with good generalization?

In this paper, we answer the above two problems positively. We propose a *BN-sharpness* to measure the sharpness for neural networks with BN. Specifically, BN-sharpness is a one dimensional integral of the change of the loss value along the "sharpest" direction in a small neighborhood while the radius of this neighborhood is scaled with the parameter magnitude. We show that BN-sharpness is scale invariant in contrast with the original definition of sharpness that usually computes the maxima of loss over a small ball of parameters around the minimizer. Our BN-sharpness is also computational efficient so that we can leverage it to regularize the training process.

To find a flat minimizer, we formulate a new optimization objective which penalize the BN-sharpness of the iterates. Due to the fact that BN-sharpness is actually a one dimensional integral, we can easily compute the sharpest direction and the gradient of BN-sharpness based on a novel approximation of the integral and the result of manifold optimization [3].

We apply our algorithm to the scenario where sharp minima arise often with bad generalization: the model trained by SGD with large batch size [12]. Our extensive experiments show preferable results over the corresponding baseline methods e.g., SGD and entropy SGD with large batches.

1.1 Related work

There are many works that discuss the relation between sharpness and generalization. [15] visualize the loss landscape of NN and find that sharp minima indeed generalize worse than flat ones. This relation is also empirically verified by the results in [12] and [24]. Moreover, [20] give a theoretical interpretation of this relation.

Similar to the problematic definition of sharpness for BN network, the definition of sharpness is proved to be ambiguous for ReLU neural network [6] due to the positive scale invariant property [19]

There are algorithms that aim to produce "flat" minima. Entropy SGD [4] uses the loss average of a cluster of points around the iterate as the objective instead of one specific point, which is motivated by the local geometry of the energy landscape. [23] argue that choosing integral of a small ball around the iterate as loss function can produce a smoother loss function and lead to a "flat" minimum. [22] use an expected loss function to produce "flat" minima and employ the PAC Bayes theory. However, all of these algorithms suffer from the complexity of high dimensional integrals.

Specifically for training neural network with batch normalization, [5] and [9] propose Riemannian approach to optimize neural network with batch normalization. However, these works do not consider employing the sharpness with batch normalization to reduce generalization error.

1.2 Contribution

(1) We show that the usual definition of sharpness: δ-sharpness [12] is ill-posed for NN with BN. We propose a scale invariant and computation-efficient BN-sharpness to measure sharpness for NN with BN.

(2) We employ BN-sharpness to regularize the training in order to achieve good generalization. We also design a computation-efficient algorithm to optimize the new BN-sharpness regularized objective and show it has preferable experimental results.

2 Background

2.1 Definitions of sharpness

Many sharpness definitions have been proposed in previous work [12, 11, 24]. Here we discuss one widely used definition: δ-sharpness, as an example.

Definition 2.1 (δ-L^p sharpness) *Let $B_2(\theta, \delta)$ be an Euclidean ball centered at θ with radius δ. Then, for a non-negative valued loss function $L(\cdot)$ and non-negative number p, the δ-L^p sharpness will be defined as*

$$S^p_{\delta\text{-}sharpness}(\theta) = \frac{\left(\int_{\theta' \in B_2(\theta,\delta)} |L(\theta') - L(\theta)|^p \, d\theta'\right)^{\frac{1}{p}}}{1 + L(\theta)}. \tag{1}$$

This is a general formula of sharpness defined as an L_p integral over a small Euclidean ball. We note that the usual δ-sharpness $S_{\delta\text{-sharpness}}$ used in [12] is actually δ-L^∞ sharpness[1]. Moreover,

[1] A detailed discussion can be referred to a long version of this paper which can be found in .

denominator of equation (1) is close to 1 when $L(\theta)$ is small i.e., θ at the minimizer. A discussion about other sharpness can be referred to Appendix B

2.2 Batch normalization

We briefly revisit the batch normalization and its properties. Batch normalization transforms the input of a linear-layer neuron from $z = \theta^T x$ to

$$BN(z) = \frac{z - E(z)}{\sqrt{Var(z)}} = \frac{\theta^T(x - E(x))}{\sqrt{\theta^T V_x \theta}}. \tag{2}$$

We can easily verify $BN(az) = BN(z)$ for any $a > 0$. Then, for a partially batch normalized neural network (Only part of neurons are batch normalized) with N parameter vector $\theta = (\theta_1, \cdots, \theta_{N_1}, \theta_{N_1+1}, \cdots, \theta_N)^T$, where θ_i is i-th parameter vector tying to a batch normalized neuron for $i = 1 \cdots N_1$, and $\theta_{N_1+1}, \cdots, \theta_N$ are parameter vectors tying with neurons without batch normalization[2]. Now we give the definition of scale transformation for partially batch normalized neural network.

Definition 2.2 *$T_{\vec{a}}(\cdot)$ denote scale transformation for a partially batch normalized neural network with*

$$T_{\vec{a}}(\theta) = (a_1\theta_1, \cdots a_{N_1}\theta_{N_1}, \theta_{N_1+1}, \cdots, \theta_N)^T, a_i > 0. \tag{3}$$

We see the loss function $L(\theta)$ satisfy $L(\theta) = L(T_{\vec{a}}(\theta))$ for any $T_{\vec{a}}(\cdot)$ [3]. We call this the property of scale invariance of BN. This property makes the original definition of sharpness problematic.

3 BN-sharpness

In this section, we propose a δ-BN sharpness to measure sharpness and discuss its properties. Before that, we first explain why the original measurement of sharpness is problematic.

Theorem 3.1 *Given a partially batch normalized network, δ-sharpness is not scale invariant.*

Proof 3.1 For $S_{\delta\text{-sharpness}}(\cdot)$, We consider a partially batch normalized neural network, without of loss generality we assume the network has batch normalized neuron in one layer. Then we choose $\vec{a}$ as $(1, \cdots, a_0, \cdots, a_0, 1, \cdots, 1)^T$ where a_0 locate in the same coordinate with batch normalized parameter. For any $\delta > 0$, we can choose

$$0 < a_0 \leq \frac{\delta}{\sqrt{N}\max_{1\leq i\leq N}\|\theta_i\|} \tag{4}$$

with $L(T_{\vec{a}}(\theta)) = L(\theta)$. Because a parameter θ with one layer zero weight matrix locate in $B_2(T_{\vec{a}}(\theta), \delta)$ which result in $\max_{\theta' \in B_2(\theta,\delta)}(L(\theta') - L(\theta))$ is at least as high as constant function. Then, the value of $S_{\delta\text{-sharpness}}(T_{\vec{a}}(\theta))$ is relatively large. ■

The above result reveals that the original definition of sharpness is not well defined for networks with BN. For example we can modify the minimizer with small generalization error to have infinite δ-sharpness if we rescale the norm of the minimizer to zero close enough. Hence, δ-sharpness is not an appropriate measurement of generalization for networks with BN due to the scale invariant property of networks with BN. Next we aim to derive a *scale invariant* sharpness for networks with BN: the δ-L^p BN-sharpness.

[2]Here we ignore the affine parameter of BN for simplicity. In fact, our theory can also applied to the BN with affine parameter as long as we integrate them into un-batch normalized parameter.

[3]For the situation of Pre-ResNet which has input $BN((\vec{1}+\theta)^T z)$ in skip connection layer. It's not a scale invariant layer, we treat these θ as un-batch normalized parameters.

Definition 3.1 (δ-L^p **BN-sharpness)** *Given a positive number δ, and a parameter point θ, and a partially batch normalized network, the δ-L^p BN-sharpness is defined as*

$$\|L(\cdot)\|_p^{\delta,\theta} = \sup_{v\in\phi(\theta)} \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left| \frac{L(\theta+tv)-L(\theta)}{\delta} \right|^p dt \right)^{\frac{1}{p}}, \tag{5}$$

where $\phi(\theta)$ is a set composed by v with $\|v_i\| = \|\theta_i\|, \sqrt{\sum_{j=N_1+1}^{N} \|v_j\|^2} = 1; i = 1, \cdots, N_1$.

We note that $v \in \phi(\theta)$ has same dimension with θ and $\phi(\theta)$ is determined by the parameter of the network. Moreover, $v \in \phi(\theta)$ has identical l_2 norm which is $\sqrt{\sum_{i=1}^{N_1} \|\theta_i\|^2 + 1}$ denoted as $\|\phi(\theta)\|$. For simplicity, we use BN-sharpness to refer to δ-L^p BN-sharpness. Some other properties of BN-Sharpness can be referred to Appendix A

Now, we prove that BN-sharpness is scale invariant. Hence, it's appropriate to measure sharpness for BN neural network.

Theorem 3.2 *The loss function $L(\theta)$ of a DNN with batch normalization satisfies $\|L(\cdot)\|_p^{\delta,\theta} = \|L(\cdot)\|_p^{\delta,T_{\vec{a}}(\theta)}$ for any $\vec{a}$ without negative element.*

Proof 3.2 For any $a \neq 0$, we notice that $L(\theta) = L(T_{\vec{a}}(\theta))$. Therefore for any v with $v \in \phi(\theta)$, we have

$$L(\theta+tv) - L(\theta) = L(T_{\vec{a}}(\theta) + tT_{\vec{a}}(v)) - L(T_{\vec{a}}(\theta)), \tag{6}$$

for $t \in [-\delta, \delta]$. It's easily to verify that

$$\int_{-\delta}^{\delta} \left| \frac{L(\theta+tv)-L(\theta)}{\delta} \right|^p dt = \int_{-\delta}^{\delta} \left| \frac{L(T_{\vec{a}}(\theta)+tT_{\vec{a}}(v))-L(T_{\vec{a}}(\theta))}{\delta} \right|^p dt$$

Since

$$\begin{aligned} \sup_{v\in\phi(\theta)} \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left| \frac{L(T_{\vec{a}}(\theta)+tT_{\vec{a}}(v))-L(T_{\vec{a}}(\theta))}{\delta} \right|^p dt \right)^{\frac{1}{p}} &= \sup_{v\in\phi(T_{\vec{a}}(\theta))} \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left| \frac{L(T_{\vec{a}}(\theta)+tv)-L(T_{\vec{a}}(\theta))}{\delta} \right|^p dt \right)^{\frac{1}{p}} \\ &= \|L(\cdot)\|_p^{\delta,T_{\vec{a}}(\theta)}, \end{aligned} \tag{7}$$

we get the conclusion. ■

The BN-sharpness is not only scale invariant but also computation efficient. As we have discussed in Section 2, the L^p norm of function $L(\cdot) - L(\theta)$ defined in $B_2(\theta, \delta)$ can be a measurement of sharpness (δ-sharpness is L^∞ norm). However, it involves a high dimensional integral ($p < \infty$) which is hard to compute in practice, which is also hold when $p = \infty$.

In contrast, BN-sharpness is a one dimensional integral along the sharpest direction v of $L(\theta)$ for each parameter component θ_i. It is computation efficient especially for the gradient which will be further discussed in Section 4. By Taylor's expansion, for each parameter component θ_i we have

$$L(\theta+tv) - L(\theta) \approx \sum_{i=1}^{N} t\nabla_\theta L(\theta)_i^T v_i + o(t\|v\|) \leq t\sum_{i=1}^{N} \|\nabla_\theta L(\theta)_i\|\|v_i\| + o(t\|\phi(\theta)\|). \tag{8}$$

Here $\nabla L(\theta)_i$ represents the gradient of $L(\cdot)$ with respect to the parameter vector θ_i. The equation (8) gives an approximation of the sharpest direction when δ is small. Accurate calculation of BN-sharpness requires optimization on the Oblique manifold which we will discuss in Section 4.1.

We also derive a relationship between BN-sharpness and generalization error. The result is based on PAC Bayesian theory [16, 17].

Theorem 3.3 *Given a "prior" distribution P(for parameter θ) over the hypothesis that is independent of the training data, with probability at least $1-\varepsilon$, we have*

$$|\mathbb{E}_u[L(\theta+u)] - \hat{L}(\theta)| \leq \delta^{1+\frac{1}{p}} \|L(\cdot)\|_p^{\delta,\theta} + 4\sqrt{\frac{1}{m}\left(KL(\theta+u||P) + \log\frac{2m}{\varepsilon}\right)}, \tag{9}$$

where $L(\cdot)$ and $\hat{L}(\cdot)$ are respectively expected loss and training loss, m is the number of training data and θ is the parameter learned from training data. As long as $u = tv$ where $t \sim U[-1, 1]$ is a uniform distribution, for any specific $v \in \delta \cdot \phi(\theta)$.

This result is obtained by adapting the equation (9) in [18] and the definition of BN-sharpness. The left hand side of equation (9) is a perturbed generalization error $|\mathbb{E}_u[L(\theta + u)] - \hat{L}(\theta)|$, where u is the perturbation variable. A small perturbation variable u will make the perturbed generalization error close to the real generalization error which involves a small δ in equation (9). However, if the norm of u tends to zero, the KL term will goes to infinity and make the bound vacuous (similar problem also exists in [20]). With a moderate δ, the sharpness related term $\delta^{1+\frac{1}{p}} \|L(\cdot)\|_p^{\delta,\theta}$ plays a role in determining the expected generation error. Hence, it causesa trade off when we chose δ under the perspective of equation (9). Even though, the result relatively gives a quantitatively description of generalization error based on BN-sharpness.

4 Regularizing Training with BN-sharpness

As we already have introduced the BN-sharpness and discussed its property, we next consider employing BN-sharpness to regularize the training process in order to find flat minima that potentially generalize well.

Specifically, we add the BN-sharpness as a regularization term to the original objective. Now the new objective becomes

$$\min_{\theta} L(\theta) + \lambda(\|L(\theta')\|_p^{\delta,\theta})^p. \tag{10}$$

An interesting property of BN-sharpness regularizer is that it does not change the minima of the original objective while it forces iterates move into a flat valley. The next theorem states that regularization term in (10) wouldn't change minima of original loss function when δ is small.

Theorem 4.1 *The minima of problem $\min_\theta L(\theta)$ are also minima of optimization problem* (10)*, when $\delta \to 0$ and $p \geq 1$.*

Proof: By Taylor expansion, we have

$$\begin{aligned}
\|L(\cdot)\|_p^{\delta,\theta} &= \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left| \frac{L(\theta + tv) - L(\theta)}{\delta} \right|^p dt \right)^{\frac{1}{p}} = \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left| \frac{tv^T \nabla_\theta L(\theta) + o(t\|\phi(\theta)\|)}{\delta} \right|^p dt \right)^{\frac{1}{p}} \\
&= \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left| \frac{(tv^T \nabla_\theta L(\theta))^p + o(t\|\phi(\theta)\|)}{\delta^p} \right| dt \right)^{\frac{1}{p}} = \frac{1}{\delta^{1+\frac{1}{p}}} \left(\int_{-\delta}^{\delta} t^p |v^T \nabla_\theta L(\theta)|^p + o(t\|\phi(\theta)\|) dt \right)^{\frac{1}{p}} \\
&= \frac{1}{\delta^{1+\frac{1}{p}}} \left(\frac{2\delta^{p+1}}{p+1} |v^T \nabla_\theta L(\theta)| + o(\delta^2 \|\phi(\theta)\|) \right)^{\frac{1}{p}} \approx \left(\frac{2}{p+1} \right)^{\frac{1}{p}} |v^T \nabla_\theta L(\theta)| (p \geq 1)
\end{aligned} \tag{11}$$

when $\delta \to 0$. Then we have

$$\lim_{\delta \to 0} \|L(\cdot)\|_p^{\delta,\theta} = \left(\frac{2}{p+1} \right)^{\frac{1}{p}} |v^T \nabla_\theta L(\theta)|. \tag{12}$$

Hence for $\lambda > 0$ and $\delta \to 0$,

$$L(\theta) + \lambda(\|L(\cdot)\|_p^{\delta,\theta})^p \geq L(\theta), \tag{13}$$

equality holds when $\nabla_\theta L(\theta) = 0$ which implies the minimums of $L(\theta)$ also minimize $L(\theta) + \lambda(\|L(\cdot)\|_p^{\delta,\theta})^p$ when δ is a small number. ■

The optimization problem (10) makes trade-off between accuracy target and flatness by adjusting λ in (10). Since we aim to find flat minima rather than flat point, an appropriate λ that achieves good balance between the two terms is crucial. We also discuss some other sharpness based algorithms in Appendix C

We next give a efficient algorithm to solve the above optimization problem (10). Here we explain the high-level idea of our algorithm. The obstacle of optimization problem (10) is calculating

the gradient of regularization term $(\|L(\theta')\|_p^{\delta,\theta})^p$ (BN-sharpness). It involves two steps: first calculating the "sharpest" direction which we need optimization on the Oblique manifold, and then computing the gradient of integral term in BN-sharpness under the "sharpest" direction.

The full algorithm is depicted in Algorithm[4] 1.

Algorithm 1 SGD with BN-sharpness regularization.

Input $\delta > 0$, even number p, initialize point $\theta^0 \in \mathbb{R}_n$, $v_0 \in \phi(\theta^0)$, regularization coefficient λ, iterations K_1, K_2 and learning rate η.
Optimization with BN-sharpness term
$k_2 = 0$
while $k_2 \leq K_2$ **and** $\nabla_\theta L(\theta^{k_2}) \neq 0$ **do**
  Search v **loop:** Iterate K_1 times to search the sharpest direction $v^{k_1}(\theta^{k_2})$ at point θ^{k_2}, return $v^{k_1}(\theta^{k_2})$;
  $\theta^{k_2+1} = \theta^{k_2} - \eta(\nabla_\theta L(\theta^{k_2}) + h(\theta^{k_2}, v^{k_1}(\theta^{k_2}), \delta, p, \lambda))$, $k_2 = k_2 + 1, v_0 \in \phi(\theta^{k_2})$
end while
return θ^{k_2}

More specifically, in Algorithm 1, the first step in the loop is calculating the sharpest direction, i.e., a process of optimization on an Oblique manifold. The second step in the loop is the gradient descent step to update parameter θ. In addition, we make an approximation to $\lambda\nabla_\theta \frac{1}{\delta}\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt$ and $\nabla_v \frac{1}{\delta}\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt$ as in equation (17) and (18) to further reduce computation load. In Algorithm 1 we denote the approximation to $\lambda\nabla_\theta \frac{1}{\delta}\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt$ as $h(\theta, v, \delta, p, \lambda)$.

4.1 Searching the Sharpest Direction v

The extra computation cost comparing to vanilla SGD is the procedure of searching the sharpest direction v, i.e., the first step in the loop in Algorithm 1.

One simple way to calculate v is directly choosing that Inspiring by equation (8), we can also choose $v_k \in \phi(\theta_k)$ with v_k^i has same direction with $\nabla L(\theta_k)_i$ for each step to avoid the searching process (Iterate K_1 times in Algorithm 1).

A more accurate way to search v is the following optimization problem:

$$\arg\max_{v\in\phi(\theta)} \frac{1}{\delta^{\frac{1}{p}}}\left(\int_{-\delta}^{\delta}\left|\frac{L(\theta+tv)-L(\theta)}{\delta}\right|^p dt\right)^{\frac{1}{p}}. \tag{14}$$

This is a constrained optimization problem and can be converted to optimization on manifold. We briefly present optimization on manifold here and detailed introduction can be referred to [3].

Optimization on manifold converts a constrained optimization problem into an un-constraint problem while iterates locate in a manifold satisfy the constraint. The optimization problem (14) defined on the general Oblique manifold. We need the related formulations in Oblique manifold to process our algorithm. A brief introduction of optimization on manifold can be referred to Appendix D

Definition 4.1 (Oblique manifold) *Oblique manifold is a subset of Euclidean space satisfy* $\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n\times p} : \mathrm{ddiag}(X^TX) = I_p\}$*, where* $\mathrm{ddiag}(\cdot)$ *is diagonal matrix of a matrix.*

Apparently, for a given θ in equation (14), v lives in a special Oblique product manifold $\|\theta\|_1\mathrm{St}(n_1,1) \times \cdots \times \|\theta\|_{N_1}\mathrm{St}(n_{N_1},1) \times \mathrm{St}(\sum_{j=N_1+1}^{N} n_j, 1)$ where n_i is the dimension of parameter θ_i.

In general, solving problem (14) needs gradient assent on manifold whose update rule is given by

$$v_{k+1} = \mathrm{Retr}_{v_k}\left(\frac{1}{L}\mathrm{grad}f(v_k)\right). \tag{15}$$

[4]Here for a vector we use the superscript to represent the iterations and subscript denote the i-th component vector.

Here $\mathrm{Rert}_{v_k}(\cdot)$ is a map from tangent space $T_{v_k}\mathcal{M}$ of point v_k to manifold $\mathcal{M}$ and $\mathrm{grad}f(v_k)$ is Riemannian gradient which is the orthogonal projection of gradient $\nabla f(v_k)$ to tangent space $T_{v_k}\mathcal{M}$. Detailed discussion of these two operators can be referred to [1].

Optimization on manifold requires Retr_x and $\mathrm{grad}f(x)$ for a $x \in \mathcal{M}$ which have the following formula for the Oblique manifold $\|\theta\|\mathrm{St}(n,1)$ respectively are

$$\begin{aligned} \mathrm{Retr}_x^{\|\theta\|}(\eta) &= \frac{x+\eta}{\|x+\eta\|}\|\theta\|, P_x^{\|\theta\|}(\eta) = \eta - \frac{x}{\|\theta\|^2}\mathrm{ddiag}(x^T\eta), \\ \mathrm{grad}f(x) &= P_x^{\|\theta\|}(\nabla f(x)) = \nabla f(x) - \frac{x}{\|\theta\|^2}\mathrm{ddiag}(x^T\nabla f(x)) \end{aligned} \tag{16}$$

where $P_x^{\|\theta\|}(\cdot)$ is projection matrix of the tangent space at x. Then we can generalize the update rule to the product of Oblique manifolds, which is gradually update v_i^k according to the manifold it locates [5].

In addition, we note that our algorithm involves $\nabla_\theta \frac{1}{\delta}\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt$ (The gradient descent for parameter θ) and $\nabla_v \frac{1}{\delta}\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt$ (Searching the sharpest direction). Accurate calculation of these two terms involves heavy computational load. Calculating the integral term requires sampling value of $L(\theta+tv)$ and $\nabla L(\theta+tv)$ and each sampling corresponds to a forward and backward process of neural network. The next proposition gives an approximation to the gradient of BN-sharpness. It reduces the number of froward and backward processes to two. The proof of this proposition can be referred to Appendix E.

Proposition 4.1 *Given $\delta > 0$, for any v satisfy $\|v\| = \|\phi(\theta)\|$, we have*

$$\begin{aligned} &\Bigg\|\lambda\nabla_\theta\frac{1}{\delta}\left(\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt\right) - \frac{\lambda}{\delta}(\nabla_\theta L(\theta)^T v)^{p-1} \\ &(\nabla_\theta L(\theta+\frac{p}{p+1}\delta v) + \nabla_\theta L(\theta-\frac{p}{p+1}\delta v) - 2\nabla_\theta L(\theta))\Bigg\| \le o(\delta\|v\|), \end{aligned} \tag{17}$$

when p is an even number.

Proposition 4.1 indicates the gradient respect to parameter θ can be replaced by a term easy to compute. On the other hand, we can achieve a similar result as

$$\begin{aligned} &\Bigg\|\nabla_v\frac{1}{\delta}\left(\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt\right) - \frac{p}{p+1}(\nabla_\theta L(\theta)^T v)^{p-1} \\ &[\nabla L_v(\theta+\frac{p+1}{p+2}\delta v) + \nabla_v L(\theta-\frac{p+1}{p+2}\delta v)]\Bigg\| \le o(\delta\|v\|), \end{aligned} \tag{18}$$

for any v. This equation gives an approximation for the gradient with respect to the vector v that locates in a product of Oblique manifolds. We denote the approximation to $\nabla_v \frac{1}{\delta}\left(\int_{-\delta}^{\delta}\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^p dt\right)$ as $g(\theta, v, \delta, p)$.

We will use the two approximation terms to replace the original gradients of BN-sharpness in our algorithm. Finally, the sharpest direction v_k of point θ in Algorithm 1 is given by

$$\begin{aligned} v_i^k &= \mathrm{Retr}_{v_i^{k-1}}^{\|\theta_i^{k-1}\|}\left(P_{v_i^{k-1}}^{\|\theta_i^{k-1}\|}(g_i(\theta, v^{k-1}, \delta, p))\right), i = 1, \cdots N_1 \\ v_*^k &= \mathrm{Retr}_{v_*^{k-1}}^{1}\left(P_{v_*^{k-1}}^{1}(g_i(\theta^{k-1}, v^{k-1}, \delta, p))\right), \end{aligned} \tag{19}$$

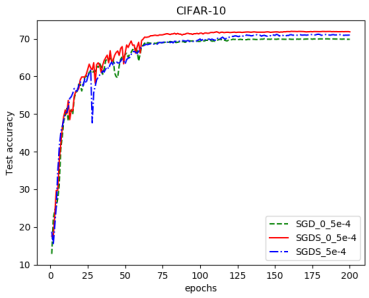
where $v_* = (v_{N_1+1}^T, \cdots, v_N^T)^T$.

5 Experiment

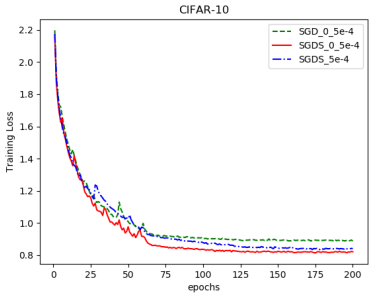
The previous work indicates that SGD with large batch size produces sharp minima [12]. Next, we use Algorithm 1 to regularize the iterates to flat minima for the case of training with large batch size. We expect our algorithm have a preferable result comparing to SGD under large batch size. We highlight that the gradient $\nabla_\theta L\left(\theta + \frac{p}{p+1}\delta v\right)$ and $\nabla_\theta L\left(\theta - \frac{p}{p+1}\delta v\right)$ in equation (17) are calculated by different batch data. It's a way of reducing variance which is used in Entropy SGD [4].

First we test the algorithm to train LeNet [14] with BN to classify CIFAR10 [13]. The hyperparameter of SGD with momentum is the following: learning rate 0.2 and multiplying by a factor 0.1 at epoch 60, 120, 160 respectively, the momentum parameter 0.9. We use batch size 10000, and weight decay 5e-4 for all the three experiments.

For experiments with regularization term, we do gradient clip with a factor 0.1 for the the gradient of BN-sharpness. We use "SGDS-number-decay" denote our algorithm with specific number of iterations of searching the sharpest direction and the specific weight decay hyperparameter. For example, "SGDS-5-5e-4" means 5 iterations of searching v in Definition 3.1 (K_1=5 in Algorithm 1) and weight decay 5e-4. For the experiments with regularized BN-sharpness, we choose $\lambda = 1e-4$ initially and increase it by a factor of 1.02 for each epoch, and choose $\delta = 0.001$ and $p = 2$. We dynamically adjust the sharpness term to avoid iterates stuck in a wide valley with low accuracy.



(a) Accuracy of LeNet on CIFAR10



(b) Training loss of LeNet on CIFAR10

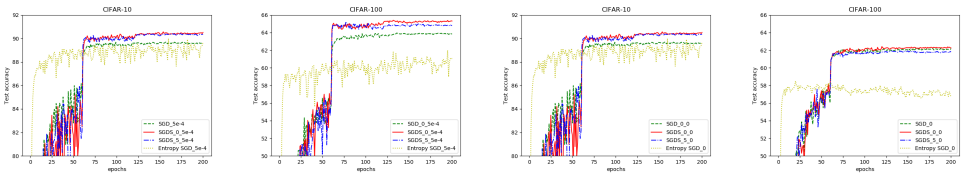
Figure 1: The performance in LeNet which are tested in the dataset of CIFAR10

The result is depicted as in Figure 1a and we can see that our algorithm has a significant improvement over vanilla SGD in this setting.

We next test our algorithm on VGG11 with bath normalization [21]. We compare the performances of vanilla SGD, SGD with BN-sharpness regularization and Entropy-SGD (denoted as E-SGD). The experiments are divided into two groups, large batch size group with batch size 2048 and small batch size group with batch size 128.

For SGDS, the δ for CIFAR10 is 5e-4 and that for CIFAR100 is 1e-3. The learning rate for SGD is 0.1 initially. All other hyperparameters are the same as those in our first experiment.

For Entropy SGD we follow the same hyperparameters in [4] for experiments with 2048 batch size, but with decaying learning rates as in SGD. For Entropy SGD with batch size 128, we turn off the dropout, use a learning rate 0.01 and keep γ constant across iterations, which produces better performance than its default setting.



(a) The accuracy of 2048 batch size VGG11 with weight decay on CIFAR10 dataset

(b) The accuracy of 2048 batch size VGG11 with weight decay on CIFAR100 dataset

(c) The accuracy of 2048 batch size VGG11 without weight decay on CIFAR10 dataset

(d) The accuracy of 2048 batch size VGG11 without weight decay on CIFAR100 dataset

Figure 2: The performance in VGG which are tested in the dataset of CIFRA10 and CIFAR100.

Algorithm	SGD-5e-4	SGDS-0-5e-4	SGDS-5-5e-4	E-SGD-5e-4
CIFAR10	91.67	91.78	91.55	91.03
CIFAR100	69.33	69.07	68.53	67.37
Algorithm	SGD	SGDS-0-0	SGDS-5-0	E-SGD-0
CIFAR10	90.5	90.53	90.53	89.96
CIFAR100	65.5	65.45	64.82	63.45

Table 1: Performance of VGG with batch size as 128.

Algorithm	SGD-5e-4	SGDS-0-5e-4	SGDS-5-5e-4	E-SGD-5e-4
CIFAR10	89.6	90.51	90.35	89.67
CIFAR100	63.83	65.31	64.8	61.02
Algorithm	SGD	SGDS-0-0	SGDS-5-0	E-SGD-0
CIFAR10	88.5	88.58	88.9	87.13
CIFAR100	62.17	62.25	61.8	56.68

Table 2: Performance of VGG with batch size as 2048.

From the Figure 2, we can see that SGD with small batch size does not encounter sharp minima, and regularizing it with BN-sharpness has little influence. Moreover, adding more iterations of searching the sharpest direction v does not make much difference. Hence we may not need to put much effort to refine the sharpest direction due to the high computation complexity.

Moreover, motivated by the claim "Train longer, generalize better" in [8], we also train a model by SGD with large batch size for 600 epochs (adjust learning rate by iterations) and observe that this model can indeed reach the result of SGDS. However, our method requires much fewer iterations that that in[8].

6 Conclusion

We first prove that original definitions of sharpness are not appropriate to describe the geometric structure of the neural network with batch normalization. Therefore, we propose a scale invariant BN-sharpness to measure the sharpness of minima. We also design an algorithm to solve the BN-sharpness regularized objective. Our algorithm does not only achieve preferable experimental result but also enjoys computational advantage in comparison to other existing sharpness based algorithms.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton Press, 2009.

[2] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:*, 2018.

[3] N. Boumal, P. A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *arxiv preprint arxiv:1605.08101*, 2017.

[4] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *In ICLR'2017 arXiv:1611.01838*, 2017.

[5] M. Cho and J. Lee. Riemannian approach to batch normalization. *arXiv preprint arXiv:1709.09603*, 2018.

[6] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. *In ICML'2017 arXiv:1703.04933*, 2017.

[7] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

[8] E. Hoffer, I. Hubara, and D. Soudryn. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[9] L. Huang, X. Liu, B. Lang, A. W. Yu, Y. Wang, and B. Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. *arXiv preprint arXiv:1709.06079*, 2017.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In Proceedings of The 32nd International Conference on Machine Learning,ICML*, pages 448–456, 2015.

[11] S. Jastrzlbski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2018.

[12] 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. *In ICLR'2017 arxiv: 1609.04836*, 2017.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278 – 2324, 1998.

[15] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. *arxiv preprint arXiv:1712.0991*, 2018.

[16] D. A. McAllesterl. Some pac-bayesian theorems. *In Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.

[17] D. A. McAllesterl. Pac-bayesian model averaging. *In Proceedings of the eleventh annual conference on Computational learning theory*, pages 164–170, 1999.

[18] D. A. McAllesterl. Simplified pac-bayesian margin bounds. *Lecture notes in computer science*, pages 203–215, 2003.

[19] Q. Meng, S. Zheng, H. Zhang, W. Chen, Z.-M. Ma, and T.-Y. Liu. G-sgd: Optimizing relu neural networks in its positively scale-invariant space. *arxiv preprint arXiv:1802.03713*, 2018.

[20] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] H. Wang, N. Keskar, C. Xiong, and R. Socher. Identifying generalization properties in neural network. *arXiv preprint arXiv:1809.07402*, 2018.

[23] W. Wen, Y. Wang, F. Yan, C. Xu, C. Wu, Y. Chen, and H. Li. Smoothout: Smoothing out sharp minima to improve generalization in deep learning. *arXiv preprint arXiv:1805.07898*, 2018.

[24] L. Wu, C. Ma, and W. E. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems 31 (NeurlPS 2018)*, 2018.

[25] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *In ICLR'2017 arXiv:1611.03530*, 2016.

A Some Properties of BN-sharpness

Though BN-sharpness is a substitution of L^p norm, we don't konw the precise relationship between δ-sharpness (L^∞ norm) and BN-sharpness. Here we theoretically analyse the BN-sharpness for a partially batch normalized neural network. Before that, we reveal mathematical meaning of δ-sharpness.

Theorem A.1 *If $L(\theta)$ is continuous, for given θ and δ, then $\max_{\theta' \in B_2(\theta,\delta)} |L(\theta') - L(\theta)| = \|L(\cdot)\|_\infty^{\delta,\theta}$ where $\|\cdot\|_\infty^{\delta,\theta}$ is the norm of measurable function space $L^\infty(B_2(\theta,\delta),\mu)$. $\|L(\cdot)\|_\infty^{\delta,\theta}$ is defined as*

$$\inf_{\mu(E_0)=0, E_0 \subset B_2(\delta,\theta)} \left(\sup_{\theta' \in B_2(\theta,\delta)-E_0} |L(\theta') - L(\theta)| \right), \tag{20}$$

where μ is Lebesgue measure.

Proof A.1 Obviously, $\max_{\theta' \in B_2(\theta,\delta)} |L(\theta') - L(\theta)| \geq \|L(\cdot)\|_\infty^{\delta,\theta}$. Suppose that

$$\inf_{\mu(E_0)=0, E_0 \subset B_2(\delta,\theta)} \left(\sup_{\theta' \in B_2(\theta,\delta)-E_0} |L(\theta') - L(\theta)| \right) = \sup_{\theta' \in B_2(\theta,\delta)-E^*} |L(\theta') - L(\theta)|,$$

and $\max_{\theta' \in B_2(\theta,\delta)} L(\theta') - L(\theta) = L(\theta^*) - L(\theta)$. If θ^* belongs to $E - E^*$, then $\max_{\theta' \in B_2(\theta,\delta)} |L(\theta') - L(\theta)| \leq \|L(\cdot)\|_\infty^{\delta,\theta}$. On the other hand, if $\theta \in E^*$, continuity of $L(\theta)$ implies that for any $\varepsilon > 0$ there exist a δ^* satisfy $|L(\theta') - L(\theta^*)| < \varepsilon$ when $\|\theta' - \theta^*\| < \delta^*$. Therefore, we have

$$|L(\theta^*) - L(\theta)| - \varepsilon < \sup_{\theta' \in B_2(\theta^*, (\delta - \|\theta^* - \theta\|) \wedge \delta^*)} |L(\theta') - L(\theta)| \leq \sup_{\theta' \in B_2(\theta,\delta)-E^*} |L(\theta') - L(\theta)|. \tag{21}$$

We conclude that $\max_{\theta' \in B_2(\theta,\delta)} |L(\theta') - L(\theta)| \geq \|L(\cdot)\|_\infty^{\delta,\theta}$ by the arbitrary of ε in (21). ■

We see that δ-sharpness is actually the L^∞ norm of function $L(\cdot) - L(\theta)$. Now, we reveal the relationship between BN-sharpness and δ-sharpness since BN-sharpness is actually a substitution of L^p norm.

Theorem A.2 *Given a continuous function $L(\theta)$, positive number δ and point θ, we have $\|L(\theta')\|_\infty^{\delta\|\phi(\theta)\|,\theta} \geq 2^{-\frac{1}{p}} \delta \|L(\theta')\|_p^{\delta,\theta}$ for any $p < \infty$, and*

$$\delta \lim_{p \to \infty} \|L(\cdot)\|_p^{\delta,\theta} = \sup_{t \in [-\delta,\delta]} \max_{v \in \phi(\theta)} L(\theta + tv) - L(\theta) \leq \|L(\cdot)\|_\infty^{\delta\|\phi(\theta)\|,\theta}. \tag{22}$$

Proof A.2 We notice that $\theta + t\delta\|\phi(\theta)\| \in B_2(\theta, \|\phi(\theta)\|)$ when $0 < t < \delta$ For any $a \neq 0$. For any v with $v \in \phi(\theta)$, we have

$$L(\theta + tv) - L(\theta) \leq \|L(\cdot)\|_{\infty}^{\delta\|\phi(\theta)\|,\theta}.$$

Then

$$\|L(\cdot)\|_p^{\delta,\theta} \leq \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left(\frac{\|L(\cdot)\|_{\infty}^{\delta\|\phi(\theta)\|,\theta}}{\delta} \right)^p dt \right)^{\frac{1}{p}} = \frac{2}{\delta}^{\frac{1}{p}} \|L(\cdot)\|_{\infty}^{\delta\|\phi(\theta)\|,\theta}, \tag{23}$$

which implies

$$\limsup_{p\to\infty} \delta\|L(\cdot)\|_p^{\delta,\theta} \leq \|L(\cdot)\|_{\infty}^{\delta\|\phi(\theta)\|,\theta}.$$

Similarly, we can confirm that

$$\limsup_{p\to\infty} \delta\|L(\cdot)\|_p^{\delta,\theta} \leq \sup_{t\in[-\delta,\delta]} \max_{v\in\phi(\theta)} L(\theta + tv) - L(\theta).$$

On the other hand, suppose $\sup_{t\in[-\delta,\delta]} \max_{v\in\phi(\theta)} L(\theta + tv) - L(\theta)$ is attained in $(t^{'}, v^{'})$. Denote $\theta + t^{'}v^{'}$ as θ^*, for any $\varepsilon > 0$, choosing v^* in BN-sharpness as

$$\left(\frac{\theta_1^* - \theta_1}{\|\theta_1^* - \theta_1\|}\|\theta_1\|, \cdots, \frac{\theta_{N_1}^* - \theta_{N_1}}{\|\theta_{N_1}^* - \theta_{N_1}\|}\|\theta_{N_1}\|, \frac{\theta_{N_1+1}^* - \theta_{N_1+1}}{\sqrt{N - N_1}\|\theta_{N_1+1}^* - \theta_{N_1+1}\|}, \cdots, \frac{\theta_N^* - \theta_N}{\sqrt{N - N_1}\|\theta_N^* - \theta_N\|} \right)^T \tag{24}$$

Let E_ε denote $\{t : |L(\theta + tv^*) - L(\theta^*)| \geq \varepsilon, -\delta \leq t \leq \delta\}$. According to the continuity of $L(\theta^{'})$, $\mu(E_\varepsilon) > 0$ where μ is one dimensional Lebesgue measure.

$$\|L(\cdot)\|_p^{\delta,\theta} \geq \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{E_\varepsilon} \left| \frac{L(\theta + tv^*) - L(\theta)}{\delta} \right|^p dt \right)^{\frac{1}{p}} \geq \left(\frac{E_\varepsilon}{\delta} \right)^{\frac{1}{p}} \frac{1}{\delta} \left(\sup_{t\in[-\delta,\delta]} \max_{v\in\phi(\theta)} L(\theta + tv) - L(\theta) - \varepsilon \right).$$

By the arbitrary of ε we conclude

$$\liminf_{p\to\infty} \delta\|L(\cdot)\|_p^{\delta,\theta} \geq \sup_{t\in[-\delta,\delta]} \max_{v\in\phi(\theta)} L(\theta + tv) - L(\theta),$$

which implies $\sup_{t\in[-\delta,\delta]} \max_{v\in\phi(\theta)} L(\theta + tv) - L(\theta) = \delta \lim_{p\to\infty} \|L(\cdot)\|_p^{\delta,\theta}$. Finally, from the definitions, we can easily conclude the last inequality in equation (22) ■

The theorem reveals that BN-sharpness is actually a weaker standard of $\delta\|\phi(\theta)\|$-sharpness.Finally, we use a theorem to reveal the influence of p in δ-L^p BN-sharpness.

Theorem A.3 *For any* $0 < p \leq q < \infty$*,* $\delta > 0$ *and* θ*, we have*

$$\|L(\cdot)\|_p^{\delta,\theta} \leq \|L(\cdot)\|_q^{\delta,\theta} \cdot (2\delta)^{\left(\frac{1}{p} - \frac{1}{q}\right)}. \tag{25}$$

Proof A.3 Suppose that

$$\|L(\cdot)\|_p^{\delta,\theta} = \left(\frac{1}{\delta} \int_{-\delta}^{\delta} \left| \frac{L(\theta + tv^*) - L(\theta)}{\delta} \right|^p \right)^{\frac{1}{p}} dt,$$

by Holder inequality and the definition of BN-Sharpness, we have

$$\begin{aligned} \left(\|L(\cdot)\|_q^{\delta,\theta}\right)^p = \left(\frac{1}{\delta} \int_{-\delta}^{\delta} \left| \frac{L(\theta + tv^*) - L(\theta)}{\delta} \right|^p \right) &\leq \left(\frac{1}{\delta} \int_{-\delta}^{\delta} \left| \frac{L(\theta + tv^*) - L(\theta)}{\delta} \right|^{p\cdot\frac{q}{p}} \right)^{\frac{p}{q}} \left(\int_{-\delta}^{\delta} 1^{\frac{q}{q-p}} dx \right)^{\frac{q-p}{q}} \\ &\leq \left(\|L(\cdot)\|_q^{\delta,\theta}\right)^{\frac{p}{q}} \cdot (2\delta)^{\frac{q-p}{q}}. \end{aligned}$$

Thus, we get the conclusion. ■

To summary, these theorems present a link between BN-Sharpness and original definition of sharpness. Specifically, BN-sharpness is actually a weaker standard of $\delta\|\phi(\theta)\|$-sharpness.Meanwhile we analysis the property of $\|L(\cdot)\|_p^{\delta,\theta}$ when $p \to \infty$. And the last property show that $\|L(\cdot)\|_p^{\delta,\theta}$ is monotone with p.

B Trace Sharpness

In addition, trace of Hessian of a minimum is another popular factor to describe flatness.

Definition B.1 (Trace-sharpness) *The trace sharpness $S_{\text{trace-sharpness}}(\theta)$ is defined as tr$(H(\theta))$, where $H(\theta)$ is the Hessian matrix on point θ.*

However, trace sharpness is also ill-posed for BN network.

Theorem B.1 *Given a partially batch normalized network, trace sharpness is not scale invariant.*

Proof B.1 For a partially batch normalized network, without of generality, we assume it is batch normalized in the first layer. We choose $\vec{a} = (a_1, 1, \cdots, 1)^T$, then we see $L(\theta) = L(T_{\vec{a}}(\theta))$ but

$$\nabla^2 L(T_{\vec{a}}(\theta)) = \begin{bmatrix} a_1^{-1} I_{n_1} & \\ & I_{n_2} \end{bmatrix} \nabla^2 l(\theta) \begin{bmatrix} a_1^{-1} I_{n_1} & \\ & I_{n_2} \end{bmatrix}, \tag{26}$$

where n_1 is the dimension of θ and n_2 is the dimension of $(\theta_2, \cdots, \theta_N)^T$. Then suppose that

$$\nabla^2 L(\theta) = \begin{bmatrix} A_1 & * \\ * & A_2 \end{bmatrix}, \tag{27}$$

where A_1, A_2 respectively have same dimension with I_{n_1}, I_{n_2}.We have

$$\text{tr}(\nabla^2 L(T_{\vec{a}}(\theta))) = a_1^{-2}\text{tr}(A_1) + \text{tr}(A_2) \neq \text{tr}(A_1) + \text{tr}(A_2) = \text{tr}(\nabla^2 L(\theta)). \tag{28}$$

It illustrates that trace-sharpness is not scale invariant. ■

We see trace sharpness can be upper bounded by δ-sharpness (by Taylor's expansion). Hence, trace sharpness is weaker than δ-sharpness. But BN-sharpness is also weaker than δ-sharpness. We use the next theorem to reveal that BN-sharpness is also a weaker standard compared to trace sharpness.

Theorem B.2 *If θ in BN-Sharpness is a minimum, then we have*

$$\|L(\cdot)\|_p^{\delta,\theta} \geq \delta \left(\frac{2}{2p+1}\right)^{\frac{1}{p}} \|\phi(\theta)\|^2 tr(H(\theta)), \tag{29}$$

where $H(\theta)$ is Hessian of θ

Proof B.2 If θ is a minimum point, then we have

$$\begin{aligned} \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left|\frac{L(\theta+tv) - L(\theta)}{\delta}\right|^p dt\right)^{\frac{1}{p}} &= \frac{1}{\delta^{\frac{1}{p}}} \left(\int_{-\delta}^{\delta} \left|\frac{t^2 v^T \nabla_\theta^2 L(\theta) v + o(t^2)}{\delta}\right|^p dt\right)^{\frac{1}{p}} \\ &\approx \frac{1}{\delta^{1+\frac{1}{p}}} \left(\int_{-\delta}^{\delta} t^{2p} \left|v^T \nabla_\theta^2 L(\theta) v\right|^p dt\right)^{\frac{1}{p}} \\ &= \delta \left(\frac{2}{2p+1}\right)^{\frac{1}{p}} v^T \nabla_\theta^2 L(\theta) v \end{aligned} \tag{30}$$

for any $v \in \phi(\theta)$. We have

$$\delta \left(\frac{2}{2p+1}\right)^{\frac{1}{p}} v^T \nabla_\theta^2 L(\theta) v \leq \delta \left(\frac{2}{2p+1}\right)^{\frac{1}{p}} \lambda_{\max}^{\theta} \|\phi(\theta)\|^2. \tag{31}$$

We have $\lambda_{\max}^{\theta} \leq \text{tr}(H(\theta))$ because $H(\theta)$ is a positive definite matrix when θ is a local minimum. Thus, we get the conclusion. ■

C Discussion About Sharpness Based Algorithms

There are plentiful sharpness based algorithms aim to producing "flat" minima. We give them a general frame. To understanding easily, we start with Entropy SGD [4]. Entropy SGD set loss function as

$$L(\theta) = -\log \int_{\theta' \in R^n} \exp\left(-f(\theta') - \frac{\gamma}{2}\|\theta' - \theta\|^2\right) d\theta', \tag{32}$$

where γ is a positive number, $f(\theta)$ is the original loss function. The $\|\theta' - \theta\|$ term in equation (32) gives more weight on the value around θ which is the motivation of deriving Entropy SGD. However, an interesting phenomenon is that the loss function of Entropy SGD is

$$L(\theta) = -\log \int_{\theta' \in R^n} \exp\left(-f(\theta') - \frac{\gamma}{2}\|\theta' - \theta\|^2\right) d\theta' \quad = -\log \mathbb{E}_{\theta'}\left[\exp(-f(\theta'))\right],$$

where $\theta' \sim \mathcal{N}(\theta, \gamma^{-\frac{1}{n}} I_n)$, n is the dimension of θ. Dividing the term in $\log$ into two parts

$$\mathbb{E}_{\theta'} \exp(-f(\theta')) = \mathbb{E}_{\theta'}\left[\exp(-f(\theta'))\mathbb{I}_{\|\theta'-\theta\|\le\delta}\right] + \mathbb{E}_{\theta'}\left[\exp(-f(\theta'))\mathbb{I}_{\|\theta'-\theta\|\ge\delta}\right]. \tag{33}$$

By Jensen inequality, we see that for any $\delta > 0$ we have

$$\begin{aligned} -\log\left(\mathbb{E}_{\theta'}\left[\exp(-f(\theta'))\mathbb{I}_{\|\theta'-\theta\|\le\delta}\right]\right) &\le -\mathbb{E}_{\theta'}\left[\mathbb{I}_{\|\theta'-\theta\|\le\delta} \log\left(\exp(-f(\theta'))\right)\right] \\ &= \mathbb{E}_{\theta'}\left[\mathbb{I}_{\|\theta'-\theta\|\le\delta} f(\theta')\right]. \end{aligned} \tag{34}$$

We conclude that the first part in equation (33) is much closer to optimization purpose. However

$$\mathbb{P}_{\theta'}(\|\theta' - \theta\| \le \delta) = \mathbb{P}_{\theta'}(\gamma^{-\frac{1}{n}}\chi(n)) = \int_0^{\gamma^{-\frac{1}{n}}\delta} \frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} dx, \tag{35}$$

which goes to zero as $n \to \infty$. That means the first term in the right hand side of equation (33) is nearly zero. Hence, loss of Entropy SGD mainly decided by the region outside a neighborhood of θ which is ambiguous to be used as a loss function.

The conclusion is counterintuitive, an intuitive explanation is that the volume of ball in n dimensional space goes to zero when n is large. Since $\log(1+x) \sim x$ when $x \to 0$, $L(\theta)$ in equation (32) is reasonable to be a loss function when $\mathbb{E}_{\theta'}\left[\exp(-f(\theta'))\mathbb{I}_{\|\theta'-\theta\|\ge\delta}\right]$ close to 1.

The general frame of algorithm based on local geometrical structure replace loss function $L(\theta)$ with

$$\int_{\mathbb{R}^n} f(\theta') d\mu_\theta(\theta'), \tag{36}$$

$\mu_\theta(\cdot)$ is a finite measure related to θ defined on $\mathbb{R}^n$, it decide the weight of point around θ contribute to loss. If $\mu_\theta(\cdot)$ is absolutely continuous to Lebesgue measure, the loss can be written as

$$\int_{\mathbb{R}^n} f(\theta') \frac{d\mu_\theta(\theta')}{d\theta'} d\theta'. \tag{37}$$

In Wen et al. [23] and Wang et al. [22], the $\frac{d\mu_\theta(\theta')}{d\theta'}$ respectively are $\mathbb{I}_{[\theta-a,\theta+a]}(\theta')$ and $\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\theta' - \theta\|^2\right)$, where a, σ are positive constants related to θ.

Under this frame, local algorithm only depends on the measure $\mu_\theta(\cdot)$ we choose. In fact, for each $\mu_\theta(\cdot)$, we can define corresponding BN-Sharpness. The only difference is substituting the norm of v in BN-Sharpness with the norm induced by $\mu_\theta(\cdot)$. In BN-Sharpness, we use Euclidean norm which induced by the Lebesgue measure constrained in a sphere. However, such a formulation has two inevitable disadvantages, the first is calculating a high dimensional integral. For a NN, the n is usually a huge number, therefore, calculating an accurate loss value of such form is difficult. The other is such loss function may appear to be inappropriate to our optimal objective, such as Entropy SGD.

D A Brief Introduction to Optimization on Manifold

A brief summary of optimization on manifold is convert the constraint condition of an optimization problem to a non-constraint problem defined on manifold. The point on such manifold always satisfy constraints. The specific definition of manifold $\mathcal{M}$ can be found in any topology book. Here we only consider matrix manifold(i.e. a subspace of Euclidean space).

Suppose $x \in \mathcal{M}$, $\mathcal{M}$ is a manifold, it has a tangent space $T_x\mathcal{M}$ which is a linear space but $\mathcal{M}$ may not. The iterates generated by gradient rise on Euclidean space is

$$x_{k+1} = x_k + \eta \nabla f(x_k), \tag{38}$$

η is step length. However, the x_k in equation (38) may not on manifold because manifold can be a nonlinear space, retraction function $\mathrm{Retr}_x(\eta) : T_x\mathcal{M} \to \mathcal{M}$ can fix this problem. Specifically, if $\mathcal{M}$ is $\mathbb{R}^n$, the Retr_x becomes $x+\eta$. We can consider η in $\mathrm{Retr}_x(\eta)$ as moving direction of iterating point. Then, the gradient ascent on manifold is given by

$$x_{k+1} = \mathrm{Retr}_x\left(\frac{1}{L}\mathrm{grad} f(x_k)\right), \tag{39}$$

where $\mathrm{grad} f(x)$ is Riemannian gradient. Riemannian gradient here is the orthogonal projection of gradient $\nabla f(x)$ to tangent space $T_x\mathcal{M}$(rigorous definition need a detailed discussion of geometry structure of manifold which is not the crucial point of this paper). Riemannian gradient is given to decide moving direction on manifold, because $\nabla f(x)$ may not in tangent space $T_x\mathcal{M}$ but the moving direction on manifold is only decided by the vector in $T_x\mathcal{M}$. All of notations related to manifold can be referred to Boumal et al.[3]. The gradient ascent on manifold helps us find the maximum of function defined on manifold. The convergence theorem of this algorithm has been proved under some extra continuous conditions [3].

E Missing Proof in Main Paper

Proof of Proposition 5.1: For any v with $v \in \phi(\theta)$, we have

$$\begin{aligned}\nabla_\theta \frac{1}{\delta}\left(\int_{-\delta}^{\delta}\left(\frac{L(\theta+te)-L(\theta)}{\delta}\right)^p dt\right) &= \frac{1}{\delta^2}\int_{-\delta}^{\delta} p\left(\frac{L(\theta+tv)-L(\theta)}{\delta}\right)^{p-1}(\nabla_\theta L(\theta+tv)-\nabla_\theta L(\theta))dt \\ &= \frac{p}{\delta^{p+1}}\int_{-\delta}^{\delta}(t\nabla_\theta L(\theta)^T v + o(t\|\phi(\theta)\|))^{p-1}(\nabla_\theta L(\theta+tv)-\nabla_\theta L(\theta))dt\end{aligned}$$

Since

$$\begin{aligned}\nabla_\theta L(\theta+tv)-\nabla_\theta L(\theta) &= \nabla_\theta L(\theta+tv)-\nabla_\theta L(\theta+\varepsilon v)+\nabla_\theta L(\theta+\varepsilon v)-\nabla_\theta L(\theta) \\ &= (t-\varepsilon)\nabla_\theta^2 L(\theta+\varepsilon v)v + o(t\|\phi(\theta)\|)+\nabla_\theta L(\theta+\varepsilon v)-\nabla_\theta L(\theta),\end{aligned}$$

let

$$h(\theta,\delta,p,v) = \left(\left(t-\frac{p}{p+1}\delta\right)\nabla_\theta^2 L\left(\theta+\frac{p}{p+1}\delta e\right)e + \nabla_\theta L\left(\theta+\frac{p}{p+1}\delta v\right)-\nabla_\theta L(\theta)\right),$$

we have

$$\begin{aligned}&\frac{p}{\delta^{p+1}}\int_{-\delta}^{0}(t\nabla_\theta L(\theta)^T v + o(t\|\phi(\theta)\|))^{p-1}(\nabla_\theta L(\theta+tv)-\nabla_\theta L(\theta))dt \\ &\qquad = \frac{p}{\delta^{p+1}}\int_{-\delta}^{0}\left(t\nabla_\theta L(\theta)^T v + o(t\|\phi(\theta)\|)\right)^{p-1}(h(\theta,\delta,p,v)+o(t\|\phi(\theta)\|))dt \\ &\qquad = \frac{1}{\delta}(\nabla_\theta L(\theta)^T v)^{p-1}\left(\nabla_\theta L\left(\theta+\frac{p}{p+1}\delta v\right)-\nabla_\theta L(\theta)\right)+o(\delta\|\phi(\theta)\|).\end{aligned}$$

Here we use the relation

$$\int_{-\delta}^{0} t^{p-1}\left(t-\frac{p}{p+1}\delta\right)dt=0.$$

Analogously, we can prove

$$\frac{p}{\delta^{p+1}}\int_{0}^{\delta}(t\nabla_\theta L(\theta)^T v+o(t\|\phi(\theta)\|))^{p-1}(\nabla_\theta L(\theta+tv)-\nabla_\theta L(\theta))dt$$
$$=\frac{1}{\delta}(\nabla_\theta L(\theta)^T v)^{p-1}\left(\nabla_\theta L\left(\theta-\frac{p}{p+1}\delta v\right)-\nabla_\theta L(\theta)\right)+o(\delta\|\phi(\theta)\|)$$

Finally, we got equation (17). ■