Information-Theoretic Local Minima Characterization and Regularization

Zhiwei Jia [1] **Hao Su** [1]

Abstract

Recent advances in deep learning theory have evoked the study of generalizability across different local minima of deep neural networks (DNNs). While current work focused on either discovering properties of good local minima or developing regularization techniques to induce good local minima, no approach exists that can tackle both problems. We achieve these two goals successfully in a unified manner. Specifically, based on the observed Fisher information we propose a metric both strongly indicative of generalizability of local minima and effectively applied as a practical regularizer. We provide theoretical analysis including a generalization bound and empirically demonstrate the success of our approach in both capturing and improving the generalizability of DNNs. Experiments are performed on CIFAR-10, CIFAR-100 and ImageNet for various network architectures.

1. Introduction

Recently, there has been a surge in the interest of acquiring a theoretical understanding over deep neural network's behavior. Breakthroughs have been made in characterizing the optimization process, showing that learning algorithms such as stochastic gradient descent (SGD) tend to end up in one of the many local minima which have close-to-zero training loss (Choromanska et al., 2015; Dauphin et al., 2014; Kawaguchi, 2016; Nguyen & Hein, 2018; Du et al., 2018). However, these numerically similar local minima typically exhibit very different behaviors in terms of generalizability. It is, therefore, natural to ask two closely related questions: (a) What kind of local minima can generalize better? (b) How to find those better local minima?

To our knowledge, existing work focused only on one of the two questions. For the "what" question, various definitions of "flatness/sharpness" have been introduced and analyzed (Keskar et al., 2017; Neyshabur et al., 2018; 2017; Wu et al., 2017; Liang et al., 2017). However, they suffer from one or more of the problems: (1) being mostly theoretical with no or poor empirical evaluations on modern neural networks, (2) lack of theoretical analysis and understanding, (3) in practice not applicable to finding better local minima. Regarding the "how" question, existing approaches (Hochreiter & Schmidhuber, 1997; Sokolić et al., 2017; Chaudhari et al., 2017; Hoffer et al., 2017; Neyshabur et al., 2015a; Izmailov et al., 2018) share some of the common drawbacks: (1) derived only from intuitions but no specific metrics provided to characterize local minima, (2) no or weak analysis of such metrics, (3) not applicable or no consistent generalization improvement for modern DNNs.

In this paper, we tackle both the "what" and the "how" questions in a unified manner. Our answer provides both the theory and applications for the generalization problems across different local minima. Based on the determinant of Fisher information estimated from the training set, we propose a metric that *solves all the aforementioned issues*. The metric can well capture properties that characterize local minima of different generalization ability. We provide its theoretical analysis, primarily a generalization bound based on PAC-Bayes (McAllester, 1999b;a). For modern DNNs in practice, it is necessary to provide a tractable approximation of our metric. We propose an intuitive and efficient approximation to compare it across different local minima. Our empirical evaluations fully illustrate the effectiveness of the metric as a strong indicator of local minima's generalizability. Moreover, from the metric we further derive and design a practical regularization technique that guides the optimization process in finding better generalizable local minima. The experiments on image classification datasets demonstrate that our approach gives consistent generalization boost for a range of DNN architectures. Codes are available at https://github.com/SeanJia/InfoMCR.

2. Related Work

It has been empirically shown that larger batch sizes lead to worse generalization (Keskar et al., 2017). Hoffer et al. (2017) analyzed how the training dynamics is affected by different batch sizes and presented a perturbed batch nor-

[1]University of California, San Diego. Correspondence to: Zhiwei Jia <zjia@ucsd.edu>, Hao Su <haosu@eng.ucsd.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

malization technique for better generalization. While it effectively improves generalization for large-batch training, a specific metric that indicates the generalizability is missing. Similarly, Elsayed et al. (2018) employed a structured margin loss to improve performance of DNNs w.r.t. noise and adversarial attack yet no metric was proposed. Furthermore, this approach essentially provided no generalization gain in the normal training setup.

The local entropy of the loss landscape was proposed to measure "flatness" in Chaudhari et al. (2017), which also designed an entropy-guided SGD that achieves faster convergence in training DNNs. However, the method does not consistently improve generalization, e.g., a decrease of performance on CIFAR-10 (Krizhevsky & Hinton, 2009). Another method that focused on modifying the optimization process is the Path-SGD proposed by Neyshabur et al. (2015a). Specifically, the authors derived an approximate steepest descent algorithm that utilizes the path-wise norm regularization to achieve better generalization. The authors only evaluated it on a two-layer neural network, very likely since the path norm is computationally expensive to optimize during training.

A flat minimum search algorithm was proposed by Hochreiter & Schmidhuber (1997) based on the "flatness" of local minima defined as the volume of local boxes. Yet since the boxes have their axes aligned to the axes of the model parameters, their volumes could be significant underestimations of "flatness" for over-parametrized networks, due to the specific spectral density of Hessian of DNNs studied in Pennington & Worah (2018); Sagun et al. (2018). The authors of Wu et al. (2017) also characterized the "flatness" by volumes. They considered the inverse volume of the basin of attraction and proposed to use the Frobenius norm of Hessian at the local minimum as a metric. In our experiments, we show that their metric does not accurately capture the generalization ability of local minima under different scenarios. Moreover, they have not derived a regularizer from their metric.

Based on a "robustness" metric, Sokolić et al. (2017) derived a regularization technique that successfully improves generalization on multiple image classification datasets. Nevertheless, we show that their metric fails to capture the generalizability across different local minima.

By using the Bayes factor, MacKay (1992) studied the generalization ability of different local minima obtained by varying the coefficient of L2 regularization. It derived a formula involving the determinant of Hessian, similar to the one in ours. Whereas, this approach has restricted settings and, without proposing an efficient approximation, its metric is not applicable to modern DNNs, let alone serving as a regularizer. A generalization bound is missing in MacKay (1992) as well.

In a broader context of the "what" question, properties that capture the generalization of neural networks have been extensively studied. Various complexity measures for DNNs have been proposed based on norm, margin, Lipschitz constant, compression and robustness (Bartlett & Mendelson, 2002; Neyshabur et al., 2015b; Sokolić et al., 2017; Xu & Mannor, 2012; Bartlett et al., 2017; Zhou et al., 2019; Dziugaite & Roy, 2017; Arora et al., 2018; Jiang et al., 2019). While some of them aimed to provide tight generalization bounds and some of them to provide better empirical results, none of the above approaches explored the "how" question at the same time.

Very recently, Karakida et al. (2019) and Sun & Nielsen (2019) studied the Fisher information of the neural network through the lens of its spectral density. In specific, Karakida et al. (2019) applied mean-field theory to study the statistics of the spectrum and the appropriate size of the learning rate. Also, an information-theoretic approach, Sun & Nielsen (2019) derived a novel formulation of the minimum description length in the context of deep learning by utilizing tools from singular semi-Riemannian geometry.

3. Outline and Notations

In a typical K-way classification setting, each sample $x \in \mathcal{X}$ belongs to a single class denoted $c_x \in \{0, 1, ..., K\}$ according to the probability vector $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the k-dimensional probability simplex so that $p(c_x = i) = y_i$ and $\sum_i y_i = 1$. Denote a feed-forward DNN parametrized by $w \in \mathbb{R}^W$ as $f_w : \mathcal{X} \to \mathcal{Y}$, which uses nonlinear activation functions and a softmax layer at the end. Denote the cross entropy loss as $\ell(f_w(x), y) = -\sum_i y_i \ln f_w(x)_i$. Denote the training set as $\mathcal{S}$, defined over $\mathcal{X} \times \mathcal{Y}$ with $|\mathcal{S}| = N$. The training objective is given as $\mathcal{L}(\mathcal{S}, w) = \frac{1}{N} \sum_{(x,y)\sim\mathcal{S}} \ell(f_w(x), y)$. Assume $\mathcal{S}$ is sampled from some true data distribution denoted $\mathcal{D}$, we can define expected loss $\mathcal{L}(\mathcal{D}, w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f_w(x), y)]$. Throughout this paper, we refer a local minimum of $\mathcal{L}(\mathcal{S}, w)$ corresponding to a local minimizer w_0 as just the local minimum w_0. Our paper's outline and main achievements are:

- In Sec. 4 we relates Fisher information to neural network training as a prerequisite.
- In Sec. 5.1 we propose a metric $\gamma(w_0)$ that well captures local minima's generalizability.
- In Sec. 5.2 we provide a generalization bound related to $\gamma(w_0)$.
- In Sec. 5.3 we propose an approximation $\widehat{\gamma}(w_0)$ for $\gamma(w_0)$, which is shown to be very effective in Sec. 7.1 via extensive empirical evaluations.
- In Sec. 6 we devise a practical regularizer from $\widehat{\gamma}(w_0)$

that consistently improves generalizability across different DNNs, as evaluated in Sec. 7.2.

3.1. Other Notations

Denote ∇_w as gradient, $\mathbf{J}_w[\cdot]$ as Jacobian matrix, ∇_w^2 as Hessian, $D_{\mathrm{KL}}(\cdot\|\cdot)$ as KL divergence, $\|\cdot\|_2$ as spectrum or Euclidean norm, $\|\cdot\|_F$ as Frobenius norm, $|\cdot|$ as determinant, $\mathrm{tr}(\cdot)$ as trace norm, $\rho(\cdot)$ as spectral radius, $\ell\ell_{\mathcal{S}}(w)$ as log-likelihood on $\mathcal{S}$, and $[\cdot]_i$ for selecting the i^{th} entry.

We define $\boldsymbol{\ell}_x(w) \in \mathbb{R}^K$ whose i^{th} entry is $-\ln f_w(x)_i$ so that $\ell(f_w(x), y) = \boldsymbol{\ell}_x(w)^T y$. We define $\tilde{y} \in \mathbb{R}^K$ as the one-hot version of y, i.e., only keep the largest dimension as 1. Then we define $\tilde{\mathcal{L}}(\mathcal{S}, w) \in \mathbb{R}^N$ as the one-hot and vectorized version of $\mathcal{L}(\mathcal{S}, w)$, i.e., a vector whose entries are $\ell(f_w(x), \tilde{y})$ for $(x, y) \in \mathcal{S}$. In other words, we approximate the cross entropy loss $\ell(f_w(x), y)$ by $\ell(f_w(x), \tilde{y})$.

4. Local Minimum and Fisher Information

First of all, if y is strictly one-hot and the training accuracy achieved at w_0 is 100%, then w_0 cannot be a local minimizer, because the cross entropy loss remains positive even if arbitrarily close to zero. To admit local minima of full training accuracy, we assume the widely used label smoothing (LS) (Szegedy et al., 2016) is applied to train all models in our analysis. LS enables us to assume a local minimum w_0 of the training loss with $\sum_{(x,y)\in\mathcal{S}} D_{\mathrm{KL}}(f_{w_0}(x)\|y) = 0$. Although empirically we find that both our proposed metric and derived regularizer work similarly well without LS.

With LS in mind, each sample $(x, y) \in \mathcal{S}$ has its label c_x sampled by $p(c_x = i|x) = y_i$, denoted as $c_x \sim y$. We denote the training data distribution as $(x, c_x) \sim \mathcal{S}$. The joint probability $p(x, c_x)$ modeled by the DNN is $p(x, c_x = i; w) = p(c_x = i|x; w)\, p(x) = [f_w(x)]_i\, p(x)$ with $p(x) = \frac{1}{N}$. We can relate the training loss $\mathcal{L}(\mathcal{S}, w)$ to the negative log-likelihood $-\ell\ell_{\mathcal{S}}(w)$ by:

$$\begin{aligned}\mathcal{L}(\mathcal{S}, w) &= \frac{1}{N}\sum_{(x,y)\in\mathcal{S}} \boldsymbol{\ell}_x(w)^T y \\ &= -\frac{1}{N}\sum_{(x,y)\in\mathcal{S}} \mathop{\mathbb{E}}_{c_x\sim y} \ln p(c_x|x; w) \\ &= -\frac{1}{N}\ell\ell_{\mathcal{S}}(w) + \ln\frac{1}{N}\end{aligned}$$

$$\text{where } -\ell\ell_{\mathcal{S}}(w) = -\sum_{(x,y)\in\mathcal{S}} \mathbb{E}_{c_x\sim y} \ln p(x, c_x; w)$$

Also, w_0 corresponds to a local maximum of the likelihood function. The observed Fisher information (Efron & Hinkley, 1978) evaluated at w_0 is defined using the Hessian of the negative log-likelihood, i.e.,

$$\begin{aligned}\mathcal{I}_{\mathcal{S}}(w_0) &= -\frac{1}{N}\nabla_w^2 \ell\ell_{\mathcal{S}}(w_0) = \nabla_w^2 \mathcal{L}(\mathcal{S}, w_0) \\ &= \mathop{\mathbb{E}}_{(x,c_x)\sim\mathcal{S}}[\nabla_w \ln p_{w_0}(c_x) \nabla_w \ln p_{w_0}(c_x)^T] \quad (1)\end{aligned}$$

where $p_{w_0}(c_x)$ denotes $p(c_x|x; w_0)$. The first equality is straightforward; the second has its proof in Appendix A. Since $p(c_x = i|x) = y_i$ and $\ln p(c_x = i|x; w_0) = [\boldsymbol{\ell}_x(w_0)]_i$, we can further simplify the Equation 1 to:

$$\mathcal{I}_{\mathcal{S}}(w_0) = \frac{1}{N}\sum_{(x,y)\in\mathcal{S}}\sum_{i=1}^{K} \nabla_w[\boldsymbol{\ell}_x(w_0)]_i \nabla_w[\boldsymbol{\ell}_x(w_0)]_i^T \quad (2)$$

Remark: A global minimum w_0, if exists, is equivalent to a local minimum with 100% training accuracy. At such w_0, we have $\nabla_w \ell(f_{w_0}(x), y) = \mathbf{0}$ as $D_{\mathrm{KL}}(f_{w_0}(x)\|y) = 0$; however, we also have $\mathcal{I}_{\mathcal{S}}(w_0) \in \mathbb{R}^{W\times W} \neq \mathbf{0}$.

5. Local Minima Characterization

In this section, we derive and propose our metric, provide a PAC-Bayes generalization bound, and lastly, propose and give intuitions of an effective approximation of our metric for modern DNNs.

5.1. Fisher Determinant as Generalization Metric

We would like a metric to compare different local minima. Under the Assumption 1, we can partition the parameter space of the neural network f_w into disjoint regions, each is a small neighborhood of a local minimum taken into account. Formally, for a local minimum w_0 and a sufficiently small $V > 0$, we define the model class $\mathcal{M}(w_0)$ as the largest connected subset of $\{w \in \mathbb{R}^W : \mathcal{L}(\mathcal{S}, w) \le h\}$ that contains w_0, where the height h is defined as a real number such that the volume (namely the Lebesgue measure) of $\mathcal{M}(w_0)$ is V. By the Intermediate Value Theorem, for any sufficiently small V there exists a corresponding height h. In essence, a local minimum w_0 of the entire parameter space becomes the global minimum of the model class $\mathcal{M}(w_0)$.

Formulated as a model class selection problem, we can compare different local minima by comparing their associated model classes. We propose our metric $\gamma(\cdot)$, where lower $\gamma(w_0)$ indicates a better generalizable local minimum w_0:

$$\gamma(w_0) = \ln|\mathcal{I}_{\mathcal{S}}(w_0)| \quad (3)$$

As a metric, $\gamma(w_0)$ requires $|\mathcal{I}_{\mathcal{S}}(w_0)| \neq 0$. Therefore, we state the following Assumption 1.

Assumption 1. *The local minima w_0 we care about in the comparison are well isolated and unique in their corresponding neighborhood $\mathcal{M}(w_0)$.*

The Assumption 1 is quite reasonable. For state-of-the-art network architectures used in practice, this is often the fact. To be precise, the Assumption 1 is violated when the Hessian matrix at a local minimum is singular. Specifically, Orhan & Pitkow (2018) summarizes three sources of the singularity: (i) due to a dead neuron, (ii) due to identical neurons, and (iii) linear dependence of the neurons. As well demonstrated in Orhan & Pitkow (2018), network with skip connection, e.g. ResNet (He et al., 2016), WRN (Zagoruyko & Komodakis, 2016), and DenseNet (Huang et al., 2017) used in our experiments, can effectively eliminate all the aforementioned singularity.

In Dinh et al. (2017), the authors pointed out another source of the singularity specifically for networks with scale-invariant activation functions, e.g. ReLU. Namely, one can rescale the model parameters layer-wise so that the underlying function represented by the network remains unchanged in the region. In practice, this issue is not critical. Firstly, most modern deep ReLU networks, e.g. ResNet, WRN, and DenseNet, have normalization layers, e.g. BatchNorm (Ioffe & Szegedy, 2015), applied before the activations. BatchNorm shifts all the inputs to the ReLU function, equivalently shifting the ReLU horizontally which makes it no longer scale-invariant. Secondly, due to the ubiquitous use of Gaussian weights initialization scheme and weight decay, most local minima obtained by gradient learning have weights of a relatively small norm. Consequently, in practice, we will not compare two local minima essentially the same but have one as the rescaled version of the other with a much larger norm of the weights.

Note that normally we have a limited size of the dataset, and so an approximation of $\gamma(w_0)$ is a must. We present our approximation scheme and its intuition in Sec. 5.3.

5.1.1. Connection to Fisher Information Approximation (FIA) Criterion

Our metric $\gamma(w_0)$ is closely related to the FIA criterion. Based on the MDL principle (Rissanen, 1978), Rissanen (1996) derived the FIA criterion to compare statistical models. Tailored to our setting, each model class $\mathcal{M}(w_0)$ has its FIA criterion as (lower FIA is better):

$$\text{FIA} = -\sum_{(x,y)\in\mathcal{S}} \mathop{\mathbb{E}}_{c_x\sim y} \ln p(x, c_x; w_0) + \frac{W}{2}\ln\frac{N}{2\pi} + \ln\int_{\mathcal{M}(w_0)} \sqrt{|\mathcal{J}(w)|}\, dw$$

Where $\mathcal{J}(w)$ is the expected Fisher information evaluated at w. Notice that all regularity conditions of the FIA criterion are satisfied for the local minimum w_0 (also the global optimum of the model class), provided 100% training accuracy and the Assumption 1. Ignoring the constant terms and assuming the training loss is locally quadratic in $\mathcal{M}(w_0)$ (later formalized and validated as Assumption 2), the RHS becomes $\ln V + \frac{1}{2}\ln|\mathcal{J}(w_0)|$. Remind that V is defined as the volume of $\mathcal{M}(w_0)$, also a constant.

Essentially in our metric we use the observed Fisher information in place of the expected one, making our metric tractable and applicable to modern DNNs.

5.1.2. Connection to Existing Flatness/Sharpness Metrics

As mentioned in Sec. 2, the "flatness" of a local minimum was firstly related to the generalization ability of the neural network in Hochreiter & Schmidhuber (1997), where the concept and the method are both preliminary. The idea is recently popularized in the context of deep learning by a series of paper such as Keskar et al. (2017); Chaudhari et al. (2017); Wu et al. (2017). Our approach roughly shares the same intuition with these existing works, namely, a "flat" local minimum admits less complexity and so generalizes better than a "sharp" one. To our best knowledge, our paper is the first among these work that provides both the theoretical analysis including a generalization bound and the empirical verification of both an efficient metric and a practical regularizer for modern network architectures.

5.2. Generalization Bound

Assumption 2. *Given the training loss $\mathcal{L}(\mathcal{S}, w)$, its local minimum w_0 satisfying Assumption 1 and the associated neighborhood $\mathcal{M}(w_0)$ whose volume V is sufficiently small, as described in Sec. 3, 4 and 5.1, respectively, when confined to $\mathcal{M}(w_0)$, we assume that $\mathcal{L}(\mathcal{S}, w)$ is quadratic.*

The Assumption 2 is quite reasonable as well. Grünwald & Grunwald (2007) suggests that, a log-likelihood function, under regularity conditions (1) existence of its 1^{st}, 2^{nd} & 3^{rd} derivatives and (2) uniqueness of its maximum in the region, behaves locally like a quadratic function around its maximum. In our case, $\mathcal{L}(\mathcal{S}, w)$ corresponds to the log-likelihood function $\ell\ell_{\mathcal{S}}(w)$ and so w_0 corresponds to a local maximum of $\ell\ell_{\mathcal{S}}(w)$. Since $\mathcal{L}(\mathcal{S}, w)$ is analytic and w_0 is the only local minimum of $\mathcal{L}(\mathcal{S}, w)$ in $\mathcal{M}(w_0)$, the training loss indeed can be considered locally quadratic.

Similar to Langford & Caruana (2002), Harvey et al. (2017) and Neyshabur et al. (2017), we apply the PAC-Bayes Theorem (McAllester, 2003) to derive a generalization bound for our metric. Specifically, we pick a uniform prior $\mathcal{P}$ over $w \in \mathcal{M}(w_0)$ according to the maximum entropy principle and pick the posterior $\mathcal{Q}$ of density $q(w) \propto e^{-|\mathcal{L}_0 - \mathcal{L}(\mathcal{S},w)|}$ with $\mathcal{L}_0 \triangleq \mathcal{L}(\mathcal{S}, w_0)$. Then Theorem 1 bounds the expected generalization loss using $\gamma(w_0)$ (proved in Appendix B).

Theorem 1. *Given $|\mathcal{S}| = N$, $\mathcal{D}$, $\mathcal{L}(\mathcal{S}, w)$ and $\mathcal{L}(\mathcal{D}, w)$ described in Sec. 3, a local minimum w_0, the volume V of $\mathcal{M}(w_0)$ sufficiently small, the Assumption 1 & 2 satisfied,*

and $\mathcal{P}, \mathcal{Q}$ defined above, for any $\delta \in (0, 1]$, we have with probability at least $1-\delta$ that:

$$\mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \leq \mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] + 2\sqrt{\frac{2\mathcal{L}_0 + 2\mathcal{A} + \ln\frac{2N}{\delta}}{N-1}}$$
$$\text{where } \mathcal{A} = \frac{1}{4\pi e} W V^{\frac{2}{W}} \pi^{\frac{1}{W}} \exp\{\frac{\boldsymbol{\gamma(w_0)}}{W}\}$$

Where W is the number of model parameters (defined in Sec. 3) and V the volume controlling the size of the neighborhood taken into account around w_0 (defined in Sec. 5.1). In short, Theorem 1 shows that a lower $\gamma(w_0)$ indicates a local minimum w_0 of better generalization.

5.3. Approximation

As stated in Sec. 4, in practice an approximation of $\gamma(w_0)$ as $\widehat{\gamma}(w_0)$ is necessary, as calculating $\gamma(w_0)$ involves computing the product of all W non-zero eigenvalues of the Fisher information matrix. Assume an imagined training set $\mathcal{S}'$ of size W and a local minimum w_0 of $\mathcal{L}(\mathcal{S}', w)$; then $\ln|\mathcal{I}_{\mathcal{S}'}(w_0)|$ is well defined on the full-rank Fisher information denoted as $\mathcal{I}_{\mathcal{S}'}(w_0)$. In reality, we only have a training set $\mathcal{S} \subset \mathcal{S}'$ with $|\mathcal{S}|$ non-zero eigenvalues of the singular matrix $\mathcal{I}_{\mathcal{S}}(w_0)$. Similar to the approach in Karakida et al. (2019), we propose to approximate eigenvalues of $\mathcal{I}_{\mathcal{S}'}(w_0)$ by the non-zero eignevalues of $\mathcal{I}_{\mathcal{S}}(w_0)$, or equivalently, as shown later, by the eigenvalues of sub-matrices of $\mathcal{I}_{\mathcal{S}'}(w_0)$.

First of all, we replace y by its one-hot version $\tilde{y}$ defined in Sec. 3.1, drastically reducing the cost of gradient calculation. This is reasonable since y and $\tilde{y}$ are very close. With $\tilde{\mathcal{L}}(\mathcal{S}, w) \in \mathbb{R}^N$ defined in Sec. 3.1, according to Equation 2, we have $\mathcal{I}_{\mathcal{S}'}(w_0) \in \mathbb{R}^{W\times W}$ as:

$$\begin{aligned}\mathcal{I}_{\mathcal{S}'}(w_0) &\approx \frac{1}{W}\sum_{(x,y)\in\mathcal{S}'} \nabla_w[\boldsymbol{\ell}_x(w_0)]_{\mathbf{y}} \nabla_w[\boldsymbol{\ell}_x(w_0)]_{\mathbf{y}}^T \\ &\quad \text{where } \mathbf{y} = \operatorname{argmax}(y) \\ &= \frac{1}{W}\mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}', w)]^T \mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}', w)] \\ &= \frac{1}{W}\mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}', w)]\, \mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}', w)]^T \end{aligned} \tag{4}$$

Let $\{\lambda_m\}_{m=1}^W$ denote the eigenvalues of $\mathcal{I}_{\mathcal{S}'}(w_0)$; then $\gamma(w_0) = \ln\prod_{m=1}^W \lambda_m = \sum_{m=1}^W \ln\lambda_m$. Ideally we want to perform a Monte-Carlo estimation of $\gamma(w_0)$ by randomly sampling $N' < N \ll W$ eigenvalues from $\{\lambda_m\}_{m=1}^W$, where N is the size of $\mathcal{S}$. We denote the samples as $\{\lambda_n\}_{n=1}^{N'}$ and we have $\frac{W}{N'}\sum_{n=1}^{N'} \ln\lambda_n \approx \sum_{m=1}^W \ln\lambda_m$. Suppose the estimation is run T times, we have $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T \frac{W}{N'}\sum_{n=1}^{N'} \ln\lambda_n = \gamma(w_0)$.

Then the eigenvalue approximation comes in. We sample $\mathcal{S}^t \subset \mathcal{S}$ i.i.d. with $|\mathcal{S}^t| = N'$ for T times and define

$$\xi^t(w_0) \triangleq \mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}^t, w_0)]\mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}^t, w_0)]^T \in \mathbb{R}^{N'\times N'} \tag{5}$$

Notice that $\xi^t(w_0)$ is a principal sub-matrix of $W\mathcal{I}_{\mathcal{S}'}(w_0)$ by removing rows & columns for data in $\mathcal{S}\setminus\mathcal{S}^t$. According to Theorem 2 in Appendix C and properties of the spectral density of Fisher information (Pennington & Worah, 2018; Sagun et al., 2018; Karakida et al., 2019), one can well approximate the eigenvalues of $\mathcal{I}_{\mathcal{S}'}(w_0)$ by those of its sub-matrices. Therefore we define the estimation $\widehat{\gamma}(w_0)$ as:

$$\widehat{\gamma}(w_0) \triangleq \frac{1}{T}\sum_{t=1}^T \ln\left|\xi^t(w_0)\right| \tag{6}$$

The relation between $\widehat{\gamma}(w_0)$ and $\gamma(w_0)$ is given as:

$$\gamma(w_0) \approx \frac{W}{N'}\widehat{\gamma}(w_0) + W\ln\frac{1}{W} \quad \text{as } T\to\infty$$

We leave the derivation of Equation 7 to Appendix C. In proposing $\widehat{\gamma}(w_0)$, we ignore the constants and irrelevant scaling factors. Empirically we find that given relatively large number of sample trials T, our metric $\widehat{\gamma}(\cdot)$ can effectively capture the generalizability of a local minimum even for a small N' (details in Sec. 7.1 and Appendix D).

6. Local Minima Regularization

Besides pragmatism, devising a practical regularizer based on $\gamma(w_0)$ also "verifies" our theoretical understanding of DNN training, helping the future improvement of the learning algorithms. Following the approximation scheme in Sec. 5.3, it is natural to regularize $\gamma(w_0)$ during mini-batch learning by minimizing the product of $|\mathcal{B}|$ non-zero eigenvalues of the Fisher information computed $\mathcal{I}_{\mathcal{B}}(w_0)$, computed via the current batch $\mathcal{B}$, other than directly minimizing $\gamma(w_0)$. However, this is far from practical due to the computation burden of:

1. computing the eigenvalues in each training step
2. computing second-order derivatives (i.e., computing the gradients of $\widehat{\gamma}(w_0)$ with respect to w_0)

There is another major challenge. All of our theoretical analysis of $\gamma(\cdot)$ works on the grounds that the Assumption 1 & 2 are reasonable and satisfied, i.e., the largest $|\mathcal{B}|$ eigenvalues of $\mathcal{I}_{\mathcal{B}}(w_0)$ evaluated at the local minimum w_0 are non-zero. However, directly minimizing the product of these positive eigenvalues pays too much attention to the smallest eigenvalues, which can easily result in zero eigenvalues, raising singularity and thus violating the assumptions. Instead, we need the effort more spread out. A good choice is to minimize the trace norm $\operatorname{tr}(\mathcal{I}_{\mathcal{B}}(w_0))$, which provides an upper bound of the product of eigenvalues in the form of:

$$\prod_i \lambda_i(\mathcal{I}_{\mathcal{B}}(w_0))^{1/|\mathcal{B}|} \leq \frac{1}{|\mathcal{B}|}\operatorname{tr}(\mathcal{I}_{\mathcal{B}}(w_0))$$

Although this bound is not be tight in general, we are tightening it when we minimize the trace norm. According to Corollary 1 in Rodin et al. (2017), we have:

$$\frac{1}{|\mathcal{B}|}\mathrm{tr}\big(\mathcal{I}_{\mathcal{B}}(w_0)\big) - \prod_i \lambda_i\big(\mathcal{I}_{\mathcal{B}}(w_0)\big)^{1/|\mathcal{B}|} \leq \sqrt{|\mathcal{B}|-1}\,\sigma$$

Where σ denotes the standard deviation of the eigenvalues of $\mathcal{I}_{\mathcal{B}}(w_0)$. As pointed out in Pennington & Worah (2018); Sagun et al. (2018); Karakida et al. (2019), these eigenvalues are highly concentrated with only a few very large "outliers" which contribute the most to the variance. When we minimize the trace norm, i.e. the L1 norm of the eigenvalues, the largest few eigenvalues bear the most weight before they are reduced to a level that has the bound effectively tightened. Furthermore, computing the trace norm does not require computing eigenvalues; thus optimizing them removes the first computation burden.

Similar to the approach in Equation 4, we approximate y by its one-hot version $\tilde{y}$. For simplicity, for the rest of this section, we denote $\tilde{y}$ as y and correspondingly denote $\mathbb{E}_j[\tilde{\mathcal{L}}(\cdot, w_0)]_j$ as $\mathcal{L}(\cdot, w_0)$, where $\tilde{\mathcal{L}}$ is the one-hot vectorized loss defined in Sec. 3.1.

Given a vector x, we have $\mathrm{tr}(xx^T) = ||x||_2^2$. Therefore, we choose to approximate the trace norm as:

$$\mathrm{tr}\big(\mathcal{I}_{\mathcal{B}}(w_0)\big) \approx \frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}} \big\|\nabla_w \ell(f_{w_0}(x), y)\big\|_2^2$$

To compute such quantity we need gradients for each individual data point. We simplify this computation by grouping data points into batches and computing averaged gradients instead. We randomly split $\mathcal{B}$ into M sub-batches of equal size, namely $\{\mathcal{B}_i\}_{i=1}^M$. We define

$$\boldsymbol{g}_i \triangleq \nabla_w \mathcal{L}(\mathcal{B}_i, w_0)$$

and then choose to optimize $\frac{1}{M}\sum_i^M \|\boldsymbol{g}_i\|_2^2$ instead of optimizing $\frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}} \|\nabla_w \ell(f_{w_0}(x), y)\|_2^2$, which drastically boosts the speed performance.

We deal with the second computation burden by adopting first order approximation. For any w, with a sufficiently small $\alpha > 0$, we have:

$$\tilde{\mathcal{L}}(\mathcal{B}_i, w - \alpha\boldsymbol{g}_i) \approx \tilde{\mathcal{L}}(\mathcal{B}_i, w) - \mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{B}_i, w)]\,\alpha\boldsymbol{g}_i$$

Thereby, we can estimate $\big\|\boldsymbol{g}_i\big\|_2^2$ by:

$$\begin{aligned}
\alpha\big\|\boldsymbol{g}_i\big\|_2^2 &= \frac{1}{|\mathcal{B}_i|}\sum_{j=1}^{|\mathcal{B}_i|}\Big[\mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{B}_i, w)]\,\alpha\boldsymbol{g}_i\Big]_j \\
&\approx \frac{1}{|\mathcal{B}_i|}\sum_{j=1}^{|\mathcal{B}_i|}[\tilde{\mathcal{L}}(\mathcal{B}_i, w) - \tilde{\mathcal{L}}(\mathcal{B}_i, w - \alpha\boldsymbol{g}_i)]_j \\
&= \mathcal{L}(\mathcal{B}_i, w) - \mathcal{L}(\mathcal{B}_i, w - \alpha\boldsymbol{g}_i)
\end{aligned}$$

Therefore, we propose to optimize the following regularized training objective for each mini-batch gradient descent step:

$$\mathcal{L}(\mathcal{B}, w) + \beta\mathcal{R}_\alpha(w) \quad \text{where} \tag{7}$$

$$\begin{aligned}
\mathcal{R}_\alpha(w) &\triangleq \frac{1}{M}\sum_{i=1}^{M}\big[\mathcal{L}(\mathcal{B}_i, w) - \mathcal{L}(\mathcal{B}_i, w - \alpha\boldsymbol{g}_i)\big] \\
&= \mathcal{L}(\mathcal{B}, w) - \frac{1}{M}\sum_{i=1}^{M}\mathcal{L}(\mathcal{B}_i, w - \alpha\boldsymbol{g}_i)
\end{aligned}$$

Illustrated in Fig. 1, an intuition is that Eq. 7 penalizes a divergent set of gradients across samples in a mini-batch.

We omit any second order term when computing $\nabla_w\mathcal{R}_\alpha(w)$, simply by not back-propagating the gradient through $\boldsymbol{g}_i$. We outline our regularized training step as Algorithm 1, which has 3 hyper-parameters: α, β and M.

Algorithm 1 Regularized Gradient Descent [1]

procedure UPDATE($w, \mathcal{B}$; α, β, M)
  $\{\mathcal{B}_i\}_{i=1}^M \leftarrow \mathcal{B}$ ▷ Split the mini-batch $\mathcal{B}$
  for $i \leftarrow 1$ to M **do**
    $\boldsymbol{g}_i \leftarrow \nabla_w\mathcal{L}(\mathcal{B}_i, w_0)$
    $\boldsymbol{g}_i \leftarrow \mathrm{copy}(\boldsymbol{g}_i)$ ▷ Stop the gradient[2]
  end for
  $\mathcal{R}_\alpha(w) \leftarrow \frac{1}{M}\sum_{i=1}^{M}\big[\mathcal{L}(\mathcal{B}_i, w) - \mathcal{L}(\mathcal{B}_i, w - \alpha\boldsymbol{g}_i)\big]$
  $\nabla_w\mathcal{L}_{\text{reg}} \leftarrow \nabla_w[\mathcal{L}(\mathcal{B}, w) + \beta\mathcal{R}_\alpha(w)]$
  Update weights w with $\nabla_w\mathcal{L}_{\text{reg}}$
end procedure

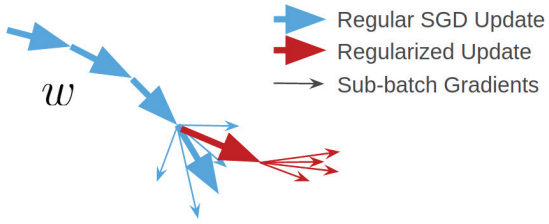

Figure 1. An illustration of Algorithm 1. In essence, the regularizer guides the optimization process to areas with less divergent gradients of different data points within a mini-batch.

7. Experiments

We perform two sets of experiments to illustrate the effectiveness of our metric $\gamma(w_0)$. We demonstrate that: (1) the approximation $\widehat{\gamma}(w_0)$ captures the generalizability well across local minima; (2) our regularization technique based on $\gamma(w_0)$ provides consistent generalization gain for DNNs.

[1] Compatible with any gradient descent-based optimizer.
[2] Implemented as `stop_gradient` in TensorFlow.

Throughout our theoretical analysis, we assume that label smoothing (LS) is applied during model training in order to obtain well-defined local minima (first mentioned in Sec. 4). In all our empirical evaluations, we perform both the version with LS applied and without. Results are very similar and so we stick to the version without LS to be consistent with the original setup in papers of the various DNNs that we used. As a result, $\tilde{y}$ and y refers to the same quantity.

7.1. Experiments on Local Minima Characterization

We perform comprehensive evaluations to compare our metric $\widehat{\gamma}(\cdot)$ with several others on ResNet-20 (He et al., 2016) for the CIFAR-10 dataset (architecture details in Appendix E). Our metric consistently outperforms others in indicating local minima's generalizability. Specifically, Sokolić et al. (2017) proposed a robustness-based metric used as a regularizer; Wu et al. (2017) proposed to use Frobenius norm of the Hessian as a metric; Keskar et al. (2017) proposed a metric closely related to the spectral radius of Hessian. In summary, we compare 4 metrics, all evaluated at a local minimum w given training set $\mathcal{S}$. All four metrics go for "smaller values indicate better generalization".

- Robustness: $\frac{1}{N}\sum_{(x,y)\in\mathcal{S}}\left\|\mathbf{J}_x[f_w(x)]\right\|_2^2$
- Frobenius norm: $\left\|\nabla_w^2\mathcal{L}(\mathcal{S},w)\right\|_F^2$
- Spectral radius: $\rho(\nabla_w^2\mathcal{L}(\mathcal{S},w))$
- Ours: $\widehat{\gamma}(w)=\frac{1}{T}\sum_{t=1}^{T}\ln|\xi(\mathcal{S}^t,w_0)|,\ \mathcal{S}^t\subset\mathcal{S}$

Both the Frobenius norm and the spectral radius based metric are related to ours, as from Equation 1 we have $\left\|\nabla_w^2\mathcal{L}(\mathcal{S},w)\right\|_F^2=\left\|\mathcal{I}_{\mathcal{S}}(w)\right\|_F^2$ and $\rho(\nabla_w^2\mathcal{L}(\mathcal{S},w))=\rho(\mathcal{I}_{\mathcal{S}}(w))$. These two metric, however, are too expensive to compute for the entire training set $\mathcal{S}$; we instead calculate them by averaging the results for T sampled $\mathcal{S}^t\subset\mathcal{S}$, similar to when we compute $\widehat{\gamma}(w)$. We leave details of how we exactly compute these metrics to Appendix D.

We perform evaluations in three scenarios, similar to Neyshabur et al. (2017); Keskar et al. (2017). We compute the 4 metrics on different local minima arising due to (1) a confusion set of varying size in training, (2) different data augmentation schemes, and (3) different batch size.

- In Scenario I, we randomly select a subset of 10000 images from CIFAR-10 as the training set and train the DNN with a confusion set consisting of images with random labels. We vary the size of the confusion set so that the resulting local minima generalize differently to the test set while all remain close-to-zero training losses. We consider confusion size of 0, 1k, 2k, 3k, 4k and 5k. We calculate all metrics based on the sampled 10000 training images.
- In Scenario II, we vary the level of data augmentation. We apply horizontal flipping, denoted `flip-only`, random cropping from images with 1 pixel padded each side plus flipping, denoted `1-crop-f`, random cropping with 4 pixels padded each side plus flipping, denoted `4-crop-f` and no data augmentation at all, denoted `no-aug`. Under all schemes, the network achieves perfect training accuracy. All the metrics are computed on the un-augmented training set.
- In Scenario III, we vary the batch size. Hoffer et al. (2017) suggests that large batch sizes lead to poor generalization. We consider the batch sizes to be 128, 256, 512 and 1024.

The default values for the 3 variables are confusion size 0, `4-crop-f` and batch size 128. For each configuration in each scenario, we train 5 models and report results (average & standard deviations) of all metrics as well as the test errors (in percentage). For the confusion set experiments, we sample a new training set and a new confusion set every time. In all scenarios, we train the model for 200 epochs with an initial learning rate 0.1, divided by 10 whenever the training loss plateaus. Within each scenario, we find the final training loss very small and very similar across different models and the training accuracy essentially equal to 1, indicating the convergence to local minima.

The results are in Figure 2, 3 and 4 for Scenario I, II and III, respectively. Our metric significantly outperforms others and is very effective in capturing the generalization properties, i.e., a lower value of our metric consistently indicates a better generalizable local minimum.

7.2. Experiments on Local Minima Regularization

We evaluate our regularizer on CIFAR-10, CIFAR-100 and the ImageNet classification task (Deng et al., 2009). For CIFAR-10 & CIFAR-100, we evaluate on four different network architectures including a plain CNN, ResNet-20, Wide ResNet (Zagoruyko & Komodakis, 2016) and DenseNet (Huang et al., 2017). We use WRN-28-2-B(3,3) from Zagoruyko & Komodakis (2016) and the DenseNet-BC-k=12 from Huang et al. (2017). We evaluate ImageNet classification on WRN-18-1.5 from Zagoruyko & Komodakis (2016). In specific, we follow Sokolić et al. (2017) to downsample all images to 128×128 and apply standard data augmentations. See Appendix E for architecture and training details. We denote the four networks as CNN, ResNet-20, WRN-28-2 / WRN-18 and DenseNet-k12, respectively.

For the three hyper-parameters α,β,M in our proposed Algorithm 1, we find α and M quite robust and manually set $\alpha=0.0001$, $M=8$ in all experiments and select β by validation via a 45k/5k training data split for each of the network architecture & dataset pair. In specific, we

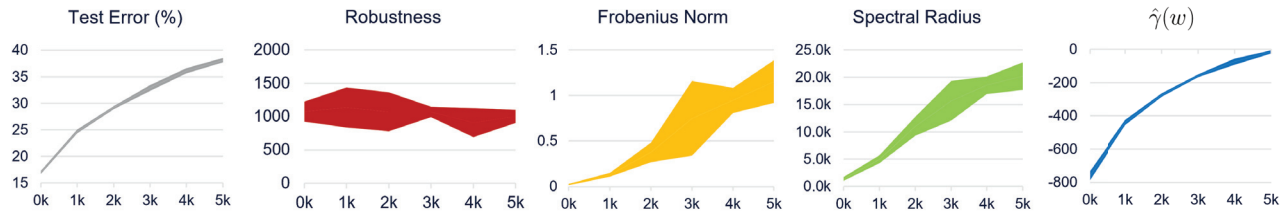

Figure 2. Scenario I: Varied size of the confusion set. 5 models are trained for each size of the confusion set (x-axis). Solid lines are the average result; shaded areas represent the $\pm$ 1 standard deviation (same for Figure 3 and 4). A larger confusion set leads to a higher test error, a trend well captured by our metric and the other two; the robustness based metric fails.

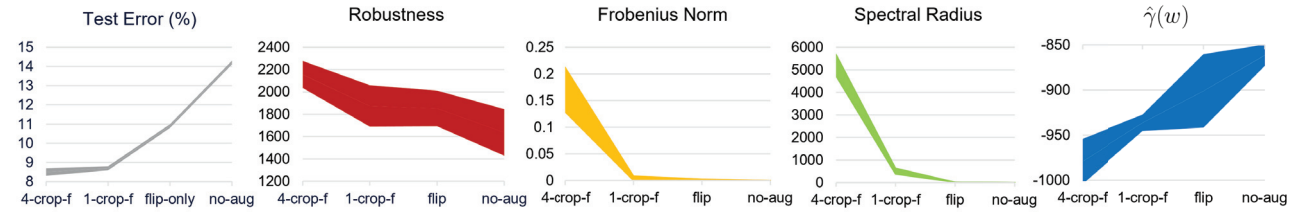

Figure 3. Scenario II: Varied data augmentation schemes. Four different schemes are used. Our metric works well as an indicator of the test error while all the other metrics completely fail.

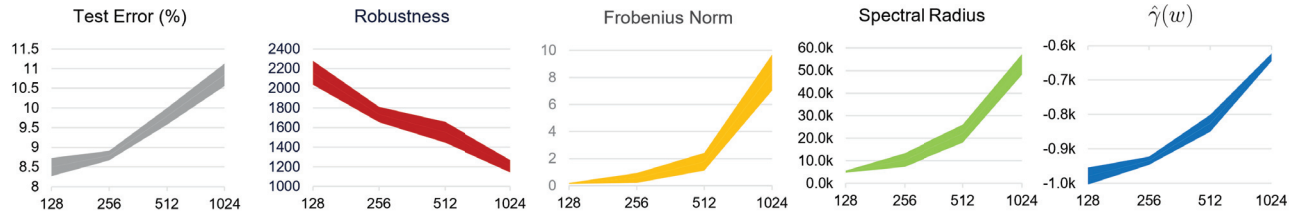

Figure 4. Scenario III: Larger batch size leads to worse generalization, captured by all the metrics except for the robustness based one.

Table 1. Test error (%) on CIFAR-10/100. In general, a model with more parameters admits more space for regularization. The representation power of ResNet-20 is too limited for CIFAR-100 (resulting in poor convergence); so we ignore it in our experiments.

	CNN	CNN+reg	WRN-28-2	WRN-28-2+reg	DenseNet-k12	DenseNet-k12+reg	ResNet-20	ResNet-20+reg
CIFAR-10	8.52 $\pm$ 0.23	**7.55 $\pm$ 0.06**	5.63 $\pm$ 0.20	**5.15 $\pm$ 0.09**	4.61 $\pm$ 0.08	**4.37 $\pm$ 0.06**	8.50 $\pm$ 0.31	**7.89 $\pm$ 0.13**
CIFAR-100	31.12 $\pm$ 0.35	**29.27 $\pm$ 0.17**	25.71 $\pm$ 0.24	**23.88 $\pm$ 0.13**	22.54 $\pm$ 0.32	**22.23 $\pm$ 0.21**	-	-

Table 2. Validation set error (%) on 128 $\times$ 128 down-sampled ImageNet classification. The better results are bolded.

Top1 Error (%)	Test	Train	Average Gap	Top5 Error (%)	Test	Train	Average Gap
WRN-18	35.52 $\pm$ 0.11	23.67 $\pm$ 2.05	11.85		14.27 $\pm$ 0.02	7.33 $\pm$ 2.25	6.94
WRN-18+reg	**34.99 $\pm$ 0.10**	**24.0 $\pm$ 3.11**	**10.99**		**13.85 $\pm$ 0.05**	**7.31 $\pm$ 1.07**	**6.54**

consider $\beta \in \{1, 5, 10, 20, 30, 40, 50, 75, 100\}$. We keep all the other training hyper-parameters, schemes as well as the setup identical to those in their original paper whenever possible (details in Appendix E). We train 5 separate models for each network-dataset combination on CIFAR-10 and CIFAR-100 and train 3 models for ImageNet. We report the test errors in percentage (mean $\pm$ std.) in Table 1 and 2, where "+reg" indicates training with our regularizer applied. The results demonstrate that our method provides consistent generalization improvement for a wide range of DNNs.

7.2.1. Time Complexity for Algorithm 1

We benchmark WRN-18 on the down-sampled ImageNet classification dataset with 2 Nvidia 2080 Ti GPUs and a batch size of 128. With parallelization, the average training time per mini-batch is 185.7ms without regularizer applied vs. 285.6ms with regularizer applied. It only takes around 1.5x longer time per gradient update for Algorithm 1.

By ablation study, we find that our regularizer works the best in the mid and late stage of DNN training, e.g., we only use the regularized update after the first learning rate drop in all of our experiments. In the beginning stage where the optimization process is not stable, our regularizer can result in great numerical errors. By only applying Algorithm 1 during the later stages, the training speed can be further increased by a large margin.

7.2.2. The Choice of The Optimizer

As described in Algorithm 1, our proposed regularizer is not tied to a specific optimizer. We perform experiments with SGD+Momentum because it is chosen to be used in ResNet, WRN, and DenseNet, helping all of them achieve current or previous state-of-the-art results. Our regularizer aims to find better "flatter" minima to improve generalization whereas adaptive optimization methods such as Adam (Kingma & Ba, 2014) and AdaGrad (Duchi et al., 2011) try to boost up convergence, yet usually at the cost of generalizability. Recent works (Wilson et al., 2017; Keskar & Socher, 2017) show that adaptive methods generalize worse than SGD+Momentum. In specific, very similar to our setup, Keskar & Socher (2017) demonstrates that SGD+Momentum consistently outperforms the others on ResNet and DenseNet for CIFAR-10 and CIFAR-100. Other approaches that also utilize local curvature to improve SGD, such as the Entropy-SGD (Chaudhari et al., 2017) mentioned in Sec. 2, have empirical results rather preliminary compared to ours.

Table 3. The proposed metric computed on local minima obtained with or without applying the proposed regularizer. Each entry represents mean $\pm$ std. among 5 runs. Smaller values are bolded.

	ResNet-20	WRN-28-2	DenseNet-k12
w/o reg.	-979.3 $\pm$ 22.3	-689.6 $\pm$ 24.9	-850.3 $\pm$ 23.5
with reg.	**-1138.1** $\pm$ **11.0**	**-748.7** $\pm$ **21.3**	**-886.2** $\pm$ **20.5**

7.2.3. Generalization Boost As a Result of Better Local Minima

We perform a sanity check to illustrate that our regularizer indeed induces better local minima characterized by our metric, i.e., our proposed regularizer is consistent with our proposed metric. For ResNet, Wide-ResNet and DenseNet trained on CIFAR-10, we compute the metric on local minima obtained with or without applying the regularizer. In specific, our regularizer has an impact on the optimization process, leaving training loss slightly different for models with or without the regularizer. To ensure our assumption that those local minima have similar close-to-zero training loss, before computing $\hat{\gamma}$ for each model, we normalize and scale the softmax output for each individual training sample. This operation makes comparison between different DNN models robust without changing their underlying behaviors. Table 3 shows that the resulting generalization boost aligns with what captured by our metric.

8. Conclusion and Future Work

In this paper, we show a bridge between the field of deep learning theory and regularization methods with respect to the generalizability of local minima. We propose a metric that captures the generalization properties of different local minima and provide its theoretical analysis including a generalization bound. We further derive an efficient approximation of the metric and a practical and effective regularizer. Empirical results demonstrate our success in both capturing and improving the generalizability of DNNs.

Moreover, we find that our proposed regularizer might be further simplified and a dynamic scheduling of the hyperparameter β can provide even more improvement to the generalization performance. In general, our exploration promises a direction for future work on the regularization and optimization of DNNs.

Acknowledgment This work was supported in part by NSF awards CNS-1730158.

References

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pp. 254–263, 2018.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1YfAfcgl.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pp. 192–204, 2015.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the

saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1019–1028. JMLR. org, 2017.

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

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Efron, B. and Hinkley, D. V. Assessing the accuracy of the maximum likelihood estimator: Observed versus expected fisher information. *Biometrika*, 65(3):457–483, 1978.

Elsayed, G., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. Large margin deep networks for classification. In *Advances in Neural Information Processing Systems*, pp. 842–852, 2018.

Grünwald, P. D. and Grunwald, A. *The minimum description length principle*. MIT press, 2007.

Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension bounds for piecewise linear neural networks. In *Conference on Learning Theory*, pp. 1064–1068, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pp. 1731–1741, 2017.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlQfnCqKX.

Karakida, R., Akaho, S., and Amari, S.-i. Universal statistics of fisher information in deep neural networks: Mean field approach. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1032–1041, 2019.

Kawaguchi, K. Deep learning without poor local minima. In *Advances in neural information processing systems*, pp. 586–594, 2016.

Keskar, N. S. and Socher, R. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Langford, J. and Caruana, R. (not) bounding the true error. In *Advances in Neural Information Processing Systems*, pp. 809–816, 2002.

Lee, C.-Y., Gallagher, P. W., and Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics*, pp. 464–472, 2016.

Liang, T., Poggio, T., Rakhlin, A., and Stokes, J. Fisher-rao metric, geometry, and complexity of neural networks. *arXiv preprint arXiv:1711.01530*, 2017.

MacKay, D. J. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

McAllester, D. Simplified pac-bayesian margin bounds. In *Learning theory and Kernel machines*, pp. 203–215. Springer, 2003.

McAllester, D. A. Pac-bayesian model averaging. In *COLT*, volume 99, pp. 164–170. Citeseer, 1999a.

McAllester, D. A. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999b.

Neyshabur, B., Salakhutdinov, R. R., and Srebro, N. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015a.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pp. 1376–1401, 2015b.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pp. 5947–5956, 2017.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

Nguyen, Q. and Hein, M. Optimization landscape and expressivity of deep cnns. In *International Conference on Machine Learning*, pp. 3727–3736, 2018.

Orhan, E. and Pitkow, X. Skip connections eliminate singularities. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkwBEMWCZ.

Pennington, J. and Worah, P. The spectrum of the fisher information matrix of a single-hidden-layer neural network. In *Advances in Neural Information Processing Systems*, pp. 5410–5419, 2018.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Rissanen, J. J. Fisher information and stochastic complexity. *IEEE transactions on information theory*, 42(1):40–47, 1996.

Rodin, B. et al. Variance and the inequality of arithmetic and geometric means. *Rocky Mountain Journal of Mathematics*, 47(2):637–648, 2017.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks, 2018. URL https://openreview.net/forum?id=rJrTwxbCb.

Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

Sun, K. and Nielsen, F. Lightlike neuromanifolds, occam's razor and deep learning. *arXiv preprint arXiv:1905.11027*, 2019.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

Wu, L., Zhu, Z., et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

Xu, H. and Mannor, S. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In Richard C. Wilson, E. R. H. and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJgqqsAct7.

Appendix

A. Proof of Equation 1 in Section 4

Let us first review the Equation 1 in Section 4:

$$\mathcal{I}_{\mathcal{S}}(w_0) = \nabla_w^2 \mathcal{L}(\mathcal{S}, w_0) = \mathop{\mathbb{E}}_{(x,c_x)\sim\mathcal{S}}[\nabla_w \ln p_{w_0}(c_x) \nabla_w \ln p_{w_0}(c_x)^T]$$

To prove this equation, it suffices to prove the following equality:

$$-\nabla_w^2 \ell\ell_{\mathcal{S}}(w) = \sum_{(x,y)\in\mathcal{S}} \sum_{i=1}^{K} y_i [\nabla_w \ln p(c_x = i|x; w) \nabla_w \ln p(c_x = i|x; w)^T]$$

For convenience, we change the notation of the local minimum from w_0 to w and further denote $p(c_x = i|x; w)$ as $p_w^x(i)$. Since $-\nabla_w^2 \ell\ell_{\mathcal{S}}(w) = -\sum_{(x,y)\in\mathcal{S}} \sum_{i=1}^{K} y_i \nabla_w^2 \ln p_w^x(i)$, for each $(x, y) \in \mathcal{S}$ and $i \in \{1, 2, ..., K\}$, we have:

$$\begin{aligned}
[\nabla_w^2 \ln p_w^x(i)]_{j,k} &= \frac{\partial^2}{\partial w_j \partial w_k} \ln p_w^x(i) \\
&= \frac{\partial}{\partial w_j} \left(\frac{\frac{\partial}{\partial w_k} p_w^x(i)}{p_w^x(i)} \right) \\
&= \frac{p_w^x(i) \frac{\partial^2}{\partial w_j \partial w_k} p_w^x(i)}{p_w^x(i)^2} - \frac{\frac{\partial}{\partial w_j} p_w^x(i)}{p_w^x(i)} \frac{\frac{\partial}{\partial w_k} p_w^x(i)}{p_w^x(i)} \\
&= \frac{\frac{\partial^2}{\partial w_j \partial w_k} p_w^x(i)}{p_w^x(i)} - \frac{\partial}{\partial w_j} \ln p_w^x(i) \cdot \frac{\partial}{\partial w_k} \ln p_w^x(i)
\end{aligned} \tag{8}$$

Since w_0 is a local minimum of full training accuracy, as described in Section 4, and $y_i = p_w^x(i)$ for $i \in \{1, 2, ..., K\}$, when taking the double summation, the first term in Equation 8 becomes:

$$\sum_{(x,y)\in\mathcal{S}} \sum_{i=1}^{K} \frac{\partial^2}{\partial w_j \partial w_k} p_w^x(i) = \frac{\partial^2}{\partial w_j \partial w_k} \sum_{(x,y)\in\mathcal{S}} \sum_{i=1}^{K} p_w^x(i) = \frac{\partial^2}{\partial w_j \partial w_k} N = 0$$

Then it follows that:

$$[\nabla_w^2 \ell\ell_{\mathcal{S}}(w)]_{j,k} = -\sum_{(x,y)\in\mathcal{S}} \sum_{i=1}^{K} y_i [\nabla_w \ln p_w^x(i) \ \nabla_w \ln p_w^x(i)^T]_{j,k}$$

B. Proof of the Generalization Bound in Section 5.2

Remind that in Section 5.2 we pick a uniform prior $\mathcal{P}$ over $w \in \mathcal{M}(w_0)$ and pick the posterior $\mathcal{Q}$ of density $q(w) \propto e^{-|\mathcal{L}_0 - \mathcal{L}(\mathcal{S},w)|}$ with $\mathcal{L}_0 \triangleq \mathcal{L}(\mathcal{S}, w_0)$. Then we have the upper bound of the expected generalization loss $\mathbb{E}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)]$ in terms of the expected training loss $\mathbb{E}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)]$ and $\gamma(w_0)$.

To prove Theorem 1, let us review the PAC-Bayes Theorem in McAllester (2003):

Theorem 2. *For any data distribution $\mathcal{D}$ and a loss function $\mathcal{L}(\cdot, \cdot) \in [0, 1]$, let $\mathcal{L}(\mathcal{D}, w)$ and $\mathcal{L}(\mathcal{S}, w)$ be the expected loss and training loss respectively for the model paramterized by w, with the training set $|\mathcal{S}| = N$. For any prior distribution $\mathcal{P}$*

with a model class $\mathcal{C}$ as its support, any posterior distribution $\mathcal{Q}$ over $\mathcal{C}$ (not necessarily Bayesian posterior), and for any $\delta \in (0, 1]$, we have with probability at least $1 - \delta$ that:

$$\mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \leq \mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] + 2\sqrt{\frac{2D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) + \ln\frac{2N}{\delta}}{N-1}}$$

PAC-Bayes (McAllester) *For a data distribution $\mathcal{D}$ and a loss $\mathcal{L}(\cdot,\cdot) \in [0, 1]$, let $\mathcal{L}(\mathcal{D}, w)$ and $\mathcal{L}(\mathcal{S}, w)$ be the expected loss and the training loss; the training set $|\mathcal{S}| = N$ is sampled from $\mathcal{D}$. Given arbitrary prior $\mathcal{P}$ and posterior $\mathcal{Q}$ (no need to be Bayesian posterior) supported on a model class $\mathcal{C}$, and for any $\delta > 0$, we have, with probability at least $1 - \delta$, that*

$$\mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \leq \mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] + 2\sqrt{\frac{2D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) + \ln\frac{2N}{\delta}}{N-1}}$$

As $e^{\gamma(w_0)} = |\mathcal{I}_{\mathcal{S}}(w_0)|$, we can rewrite the generalization bound we want to prove above as:

$$\mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \leq \mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] + 2\sqrt{\frac{W \cdot V^{2/W}\pi^{1/W}\left|\mathcal{I}_{\mathcal{S}}(w_0)\right|^{1/W} + 4\pi e\mathcal{L}_0 + 2\pi e\ln\frac{2N}{\delta}}{2\pi e(N-1)}}$$

As defined in Section 5.2, given the model class $\mathcal{M}(w_0)$, whose volume is V, for the neural network f_w, the uniform prior $\mathcal{P}$ attains the probability density function $p(w) = \frac{1}{V}$ for any $w \in \mathcal{M}(w_0)$ and the posterior $\mathcal{Q}$ has density $q(w) \propto e^{-|\mathcal{L}(\mathcal{S},w)-\mathcal{L}_0|}$. Based on Assumption 2 in Section 5.2 and the observed Fisher information $\mathcal{I}_{\mathcal{S}}(w_0)$, especially the Equation 2 derived in Section 4, we have:

$$\mathcal{L}(\mathcal{S}, w) = \mathcal{L}_0 + \frac{1}{2}(w - w_0)^T\mathcal{I}_{\mathcal{S}}(w_0)(w - w_0) \quad \forall w \in \mathcal{M}(w_0)$$

Denote $\Sigma = [\mathcal{I}_{\mathcal{S}}(w_0)]^{-1} = [\nabla^2_w\mathcal{L}(\mathcal{S}, w_0)]^{-1}$. Then $\mathcal{Q}$ is a truncated multivariate Gaussian distribution whose density function q is:

$$\begin{aligned} q(w; w_0, \Sigma) &= \frac{\sqrt{(2\pi)^{-n}|\Sigma|^{-1}}\exp\{-\frac{1}{2}(w - w_0)^T\Sigma^{-1}(w - w_0)\}}{\int_{\mathcal{M}(w_0)}\sqrt{(2\pi)^{-n}|\Sigma|^{-1}}\exp\{-\frac{1}{2}(w - w_0)^T\Sigma^{-1}(w - w_0)\}\,dw} \\ &= \frac{\exp\{-\frac{1}{2}(w - w_0)^T\Sigma^{-1}(w - w_0)\}}{\int_{\mathcal{M}(w_0)}\exp\{-\frac{1}{2}(w - w_0)^T\Sigma^{-1}(w - w_0)\}\,dw} \end{aligned} \tag{9}$$

Denote the denominator of Equation 9 as $\mathbf{Z}$ and define:

$$g(w; w_0, \Sigma) \triangleq -\frac{1}{2}(w - w_0)^T\Sigma^{-1}(w - w_0)\} \leq 0$$

Then q can also be written as:

$$q(w; w_0, \Sigma) = \frac{\exp\{g(w; w_0, \Sigma)\}}{\mathbf{Z}}$$

In order to derive a generalization bound in the form of the PAC-Bayes Theorem, it suffices to prove an upper bound of the

KL divergence term:

$$\begin{aligned}
D_{\mathrm{KL}}(\mathcal{Q}||\mathcal{P}) &= \mathop{\mathbb{E}}_{w\sim\mathcal{Q}} \ln \frac{q(w)}{p(w)} \\
&= -\mathop{\mathbb{E}}_{w\sim\mathcal{Q}} \ln \frac{1}{V} + \mathop{\mathbb{E}}_{w\sim\mathcal{Q}} \ln q(w) \\
&= \ln V + \mathop{\mathbb{E}}_{w\sim\mathcal{Q}} g(w; w_0, \Sigma) + \ln \frac{1}{\mathbf{Z}} \\
&\leq \ln V + \mathop{\mathbb{E}}_{w\sim\mathcal{Q}} 0 - \ln \left(\int_{\mathcal{M}(w_0)} \exp\{g(w; w_0, \Sigma)\}\, dw \right) \\
&\leq \ln V - \ln \left(\int_{\mathcal{M}(w_0)} \exp\{-\max_{w\in\mathcal{M}(w_0)} \mathcal{L}(\mathcal{S}, w)\}\, dw \right) \\
&= \ln V - \ln \left(V \cdot \exp\{-\max_{w\in\mathcal{M}(w_0)} \mathcal{L}(\mathcal{S}, w)\} \right) \\
&= \ln V - \ln V + h \quad = \quad h
\end{aligned}$$

where h is the height of $\mathcal{M}(w_0)$ defined in Section 5.1. For convenience, we shift down $\mathcal{L}(\mathcal{S}, w)$ by $\mathcal{L}_0$ and denote the shifted training loss $\mathcal{L}_0(w) \triangleq \mathcal{L}(\mathcal{S}, w) - \mathcal{L}_0$ so that $\mathcal{L}_0(w_0) = 0$. Then

$$\mathcal{L}_0(w) = \frac{1}{2}(w - w_0)^T \Sigma^{-1} (w - w_0) \quad \forall w \in \mathcal{M}(w_0)$$

Furthermore, the following two sets are equivalent

$$\{w \in \mathbb{R}^W : \mathcal{L}(\mathcal{S}, w) = h\} = \{w \in \mathbb{R}^W : \mathcal{L}_0(w) = h - \mathcal{L}_0\}$$

both of which are the W-dimensional hyperellipsoid given by the equation $\mathcal{L}_0(w) = h - \mathcal{L}_0$, which can be converted to the standard form for hyperellipsoids as:

$$(w - w_0)^T \frac{\Sigma^{-1}}{2(h - \mathcal{L}_0)} (w - w_0) = 1$$

The volume enclosed by this hyperellipsoid is exactly the volume of $\mathcal{M}(w_0)$, i.e., V; so we have

$$\frac{\pi^{W/2}}{\Gamma(\frac{W}{2} + 1)} \sqrt{2^W (h - \mathcal{L}_0)^W |\Sigma|} = V$$

Solve for h, with the Stirling's approximation for factorial $\Gamma(n+1) \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$, we have

$$h = \mathcal{L}_0 + \frac{\left(V \cdot \Gamma(\frac{W}{2} + 1)\right)^{2/W}}{2\pi \left|\Sigma\right|^{1/W}} \approx \mathcal{L}_0 + \frac{V^{2/W} \pi^{1/W} W^{(W+1)/W} \left|\mathcal{I}_{\mathcal{S}}(w_0)\right|^{1/W}}{4\pi e}$$

where $\Gamma(\cdot)$ denotes the Gamma function. Notice that for modern DNNs we have $W \gg 1$, and so $W^{\frac{W+1}{W}} \approx W$. We finally can derive the generalization bound in the form of the PAC-Bayes Theorem as:

$$\mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \leq \mathop{\mathbb{E}}_{w\sim\mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] + 2\sqrt{\frac{W \cdot V^{2/W} \pi^{1/W} \left|\mathcal{I}_{\mathcal{S}}(w_0)\right|^{1/W} + 4\pi e \mathcal{L}_0 + 2\pi e \ln \frac{2N}{\delta}}{2\pi e (N-1)}}$$

C. Derivation of Equation 6 in Section 5.3

First, let us present the well-known theorem in linear algebra that relates the eigenvalues of a matrix to those of its sub-matrices.

Theorem 3. *Given an $n \times n$ real symmetric matrix A with eigenvalues $\lambda_1 \leq ... \leq \lambda_n$, for any $k < n$ denote its principal sub-matrix as B obtained from removing $n-k$ rows and columns from A. Let $\nu_1 \leq ... \leq \nu_k$ be the eigenvalues of B. Then for any $1 \leq r \leq k$, we have $\lambda_r \leq \nu_r \leq \lambda_{r+n-k}$.*

Let $\{\nu_n\}_{n=1}^{N'}$ be the eigenvalues of $\frac{1}{W}\xi^t(w_0)$, which is a $N' \times N'$ sub-matrix of $\mathcal{I}_{\mathcal{S}'}(w_0)$; then

$$\widehat{\gamma}(w_0) = \frac{1}{T}\sum_{t=1}^{T} \ln\left|\xi^t(w_0)\right| = \frac{1}{T}\sum_{t=1}^{T} \ln\left|W \cdot \frac{1}{W}\xi^t(w_0)\right| = N' \ln W + \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N'} \ln \nu_n$$

Theorem 3 gives the relation between ν_n and λ_n, defined above and in Section 5.3 as the n^{th} smallest eigenvalues of $\frac{1}{W}\xi^t(w_0)$ and that of $\mathcal{I}_{\mathcal{S}'}(w_0)$, respectively. For sufficiently large N', we can use ν_n to approximate λ_n, which ignores the eigenvalues of $\mathcal{I}_{\mathcal{S}'}(w_0)$ larger than $\lambda_{N'}$. This is reasonable when estimating $\gamma(w_0)$, since in general the majority of the eigenvalues of the Hessian for DNNs are close to zero with only a few large "outliers", and so the smallest eigenvalues are the dominant terms in $\gamma(w_0)$ (Pennington & Worah, 2018; Sagun et al., 2018; Karakida et al., 2019). A specific bound of the eigenvalues remains an open question, though. In short, we have $\sum_{n=1}^{N'} \nu_n \approx \sum_{n=1}^{N'} \lambda'_n$ and consequently:

$$\begin{aligned}
\frac{W}{N'}\widehat{\gamma}(w_0) + W \ln \frac{1}{W} &= \frac{W}{N'}\widehat{\gamma}(w_0) - W \ln W \\
&= \frac{W}{N'}\Big(\widehat{\gamma}(w_0) - N' \ln W\Big) \\
&= \frac{1}{T}\sum_{t=1}^{T}\frac{W}{N'}\sum_{n=1}^{N'} \ln \nu_n \\
&\approx \frac{1}{T}\sum_{t=1}^{T}\frac{W}{N'}\sum_{n=1}^{N'} \ln \lambda'_n
\end{aligned}$$

Finally we we have

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\frac{W}{N'}\sum_{n=1}^{N'} \ln \lambda'_n = \gamma(w_0)$$

D. Details of Calculating the Metrics in Section 7.1

For the following three metrics, we apply estimation by sampling a subset $\mathcal{S}^t$ from the full training set $\mathcal{S}$ for T times and averaging the results.

- Frobenius norm: $\left\|\nabla_w^2 \mathcal{L}(\mathcal{S}, w)\right\|_F^2$
- Spectral radius: $\rho(\nabla_w^2 \mathcal{L}(\mathcal{S}, w))$
- Ours: $\widehat{\gamma}(w) = \frac{1}{T}\sum_{t=1}^{T} \ln|\xi(\mathcal{S}^t, w_0)|$

For the Frobenius norm based metric, from Equation 1 & 2 in Section 4 we have:

$$\left\|\nabla_w^2 \mathcal{L}(\mathcal{S}, w)\right\|_F^2 = \left\|\mathcal{I}_{\mathcal{S}}(w)\right\|_F^2 = \frac{1}{N}\sum_{(x,y)\in\mathcal{S}}\sum_{i=1}^{K}\left\|\left(\nabla_w[\boldsymbol{\ell}_x(w_0)]_i\right)\left(\nabla_w[\boldsymbol{\ell}_x(w_0)]_i\right)^T\right\|_F^2$$

We define $\mathbf{y} = \arg\max(y)$. Similar to Equation 4 in Section 5.3, we approximate y by $\tilde{y}$ and so

$$\left\|\nabla_w^2 \mathcal{L}(\mathcal{S}, w)\right\|_F^2 \approx \frac{1}{N}\sum_{(x,y)\in\mathcal{S}}\left\|\left(\nabla_w[\boldsymbol{\ell}_x(w_0)]_{\mathbf{y}}\right)\left(\nabla_w[\boldsymbol{\ell}_x(w_0)]_{\mathbf{y}}\right)^T\right\|_F^2$$

Summing over the entire Hessian matrix is too expensive as there are $W \times W \times N$ entries in total. We therefore estimate the quantity by first sampling a subset $\mathcal{S}^t \subset \mathcal{S}$ and then sampling 100,000 entries of $\left(\nabla_w[\boldsymbol{\ell}_x(w_0)]_{\mathbf{y}}\right)\left(\nabla_w[\boldsymbol{\ell}_x(w_0)]_{\mathbf{y}}\right)^T$. We perform the estimation T times and average the results, similar to the approach when computing $\widehat{\gamma}(w)$.

Also by Equation 2 and the approximation in Equation 4, the spectral radius of Hessian is equivalent to the squared spectral norm of $1/\sqrt{N}\mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}, w)]$. We also perform estimation (with irrelevant scaling constants dropped) by sampling $\mathcal{S}^t$ for T times, i.e., via $\frac{1}{T}\sum_t \left\|\mathbf{J}_w[\tilde{\mathcal{L}}(\mathcal{S}^t, w)]\right\|_2^2$.

Furthermore, in all our experiments that involves samplings $\mathcal{S}^t$, we set $|\mathcal{S}^t| = N' = T = 100$.

E. Architecture And Training Details in Section 7

Architecture details are as below

- The plain CNN is a 6-layer convolutional neural network similar to the baseline in Lee et al. (2016) yet without the "mlpconv" layers (resulting in a much fewer number of parameters). Specifically, the 6 layers has numbers of filters as $\{64, 64, 128, 128, 192, 192\}$. We use 3×3 kernel size and ReLU as the activation function. After the second and the fourth convolutional layer we insert a 2×2 max pooling operation. After the last convolutional layer, we apply a global average pooling before the final softmax classifier.
- For ResNet-20, WRN-28-2-B(3,3), WRN-18-1.5 and DenseNet-BC-k=12, we use the same architecture as in their original papers, respectively.

The training details are

- For the plain CNN, we initialize the weights according to the scheme in He et al. (2016) and apply l2 regularization of a coefficient 0.0001. We perform standard data augmentation, the one denoted `4-crop-f` in Section 7.1. We use stochastic gradient descent with Nesterov momentum set to 0.9 and a batch size of 128. We train 200 epochs in total with the learning rate initially set to 0.01 and then divided by 10 at epoch 100 and 150.
- For ResNet-20, WRN-28-2-B(3,3), WRN-18-1.5 and DenseNet-BC-k=12, we use the same hyper-parameters, training schemes, data augmentation schemes, optimization methods, etc., as those in their original papers, respectively. An exception is that for WRN-18-1.5 on ImageNet, we first resize all training images to 128×128, and then apply random crop (of size 114×114), horizontal flip and standard color jittering together with mean channels subtraction as in He et al. (2016). We adopt single crop (central crop) testing for the down-sampled 128×128 validation images.