Identifying Generalization Properties in Neural Networks

Huan Wang, Nitish Shirish Keskar, Caiming Xiong, Richard Socher
Salesforce Research
{huan.wang, nkeskar, cxiong, rsocher}@salesforce.com

Editor:

Abstract

While it has not yet been proven, empirical evidence suggests that model generalization is related to local properties of the optima which can be described via the Hessian. We connect model generalization with the local property of a solution under the PAC-Bayes paradigm. In particular, we prove that model generalization ability is related to the Hessian, the higher-order "smoothness" terms characterized by the Lipschitz constant of the Hessian, and the scales of the parameters. Guided by the proof, we propose a metric to score the generalization capability of the model, as well as an algorithm that optimizes the perturbed model accordingly.

Keywords: generalization, PAC-Bayes, Hessian, perturbation.

1. Introduction

Deep models have proven to work well in applications such as computer vision (Krizhevsky et al., 2012) (He et al., 2014) (Karpathy et al., 2014), speech recognition (Mohamed et al., 2012) (Hinton et al., 2012), and natural language processing (Socher et al., 2013) (Graves, 2013) (McCann et al., 2018). Many deep models have millions of parameters, which is more than the number of training samples, but the models still generalize well (Huang et al., 2017).

On the other hand, classical learning theory suggests the model generalization capability is closely related to the "complexity" of the hypothesis space. This seems to be a contradiction to the empirical observations that over-parameterized models generalize well on the test data. Indeed, even if the hypothesis space is complex, the final solution learned from a given training set may still be simple. An example is, suppose the hypothesis space is the union of linear classifiers and some complex function spaces. As a union set the hypothesis space is complex in the worst case, but for some training set the best solution may be a linear classifier. This suggests the generalization capability of the model is also related to the property of the solution.

Keskar et al. (2016) and Chaudhari et al. (2016) empirically observe that the generalization ability of a model is related to the spectrum of the Hessian matrix $\nabla^2 L(w^*)$ evaluated at the solution, and large eigenvalues of the $\nabla^2 L(w^*)$ often leads to poor model generalization. Also, (Keskar et al., 2016), (Chaudhari et al., 2016) and (Novak et al., 2018b) introduce several different metrics to measure the "sharpness" of the solution, and demonstrate the connection between the sharpness metric and the generalization empirically. Dinh et al. (2017) later points out that most of the Hessian-based sharpness measures are problematic and cannot be applied directly to explain generalization. In particular, they show that the geometry of the parameters in RELU-MLP can be modified drastically by re-parameterization.

Another line of work originates from the theorists. (Langford and Caruana, 2001) and more recently (Harvey et al., 2017) (Neyshabur et al., 2017a) (Neyshabur et al., 2017b) use PAC-Bayes bound to analysis the generalization behavior of the deep models. Since the PAC-Bayes bound holds uniformly for all "posteriors", it also holds for some particular "posteriors", for example, the solution parameter perturbed with noise. This provides a natural way to incorporate the local property of the solution into the generalization analysis. In particular, Neyshabur et al. (2017a) suggests to use the difference between the perturbed loss and the empirical loss as the sharpness metric. Dziugaite and Roy (2017) tries to optimize the PAC-Bayes bound instead for a better model generalization. Still some fundamental questions remain unanswered. In particular we are interested in the following question:

How is model generalization related to local "smoothness" of a solution?

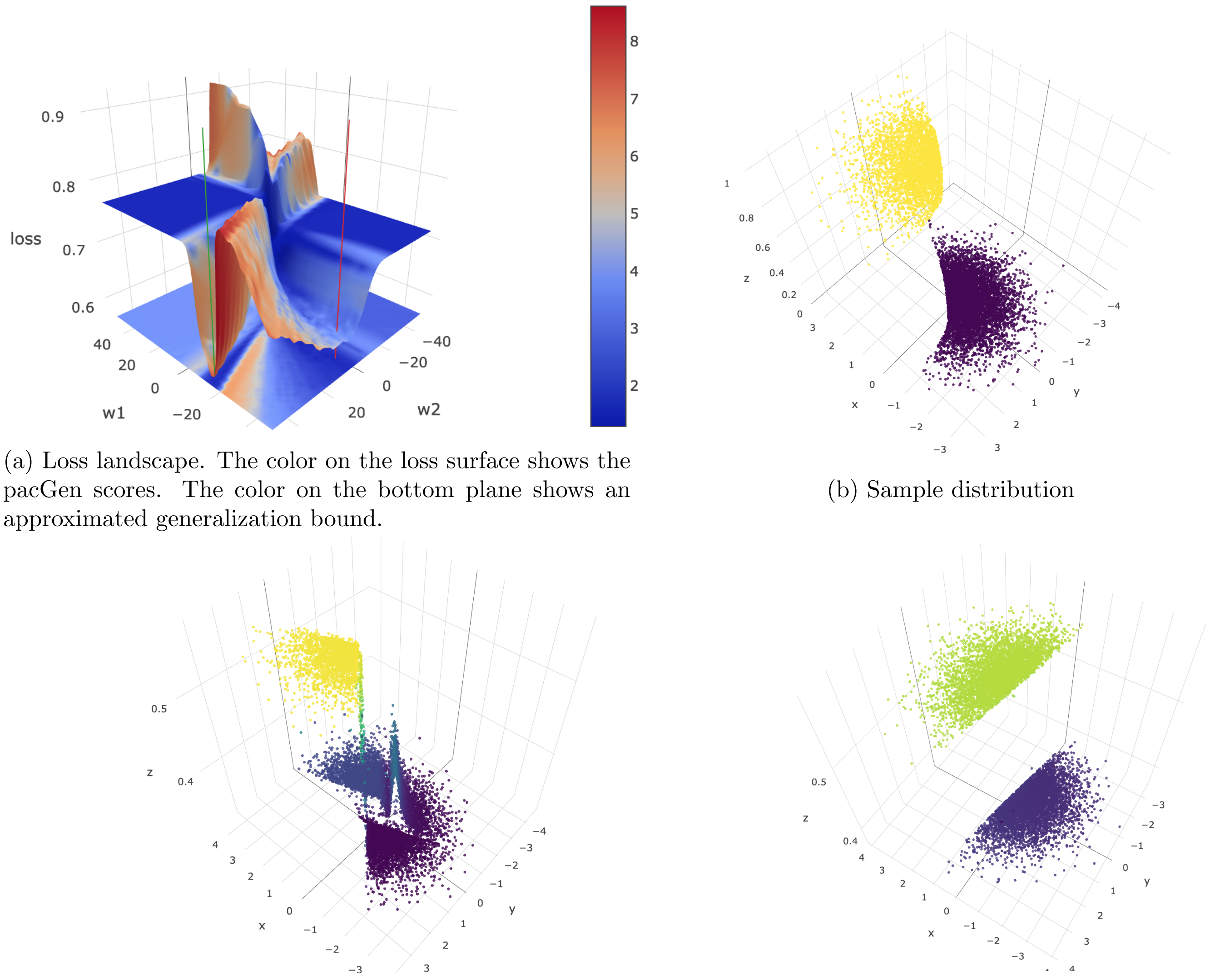
In this paper we try to answer the question from the PAC-Bayes perspective. Under mild assumptions on the Hessian of the loss function, we prove the generalization error of the model is related to this Hessian, the Lipschitz constant of the Hessian, the scales of the parameters, as well as the number of training samples. The analysis also gives rise to a new metric for generalization. Based on this, we can approximately select an optimal perturbation level to aid generalization which interestingly turns out to be related to Hessian as well. Inspired by this observation, we propose a perturbation based algorithm that makes use of the estimation of the Hessian to improve model generalization.

2. Sharp Minimum v.s. Flat Minimum - A Toy Example

Let us start with a toy example to demonstrate different behaviors of local optima. For training, we construct a small 2-dimensional sample set from a mixture of 3 Gaussians, and then binarize the labels by thresholding them from their median value. The sample distribution is shown in Figure 1b. Then we use a 5-layer MLP model with sigmoid as the activation and cross entropy as the loss for training and prediction. The variables from different layers are shared so that the model only has two free parameters w_1 and w_2.

The model is trained using 100 samples. Fixing the samples, we plot the loss function with respect to the model variables $\hat{L}(w_1, w_2)$, as shown in Figure 1a. Many local optima are observed even in this simple two-dimensional toy example. In particular a sharp one, marked by the vertical green line, and a flat one, marked by the vertical red line. The colors on the loss surface display the values of the generalization metric scores (pacGen), which we will define in section 7. Smaller metric value indicates better generalization power.

As displayed in the figure, the metric score around the global optimum, indicated by the vertical green bar, is high, suggesting possible poor generalization capability as compared to the local optimum indicated by the red bar. We also plot a plane on the bottom of the figure. The color projected on the bottom plane indicates an approximated generalization



(a) Loss landscape. The color on the loss surface shows the pacGen scores. The color on the bottom plane shows an approximated generalization bound.

(b) Sample distribution

(c) Predicted labels by the sharp minimum

(d) Predicted labels by the flat minimum

Figure 1: Loss Landscape and Predicted Labels of a 5-layer MLP with 2 parameters.

bound, which considers both the loss and the generalization metric.[1] The local optimum indicated by the red bar, though has a slightly higher loss, has a similar overall bound compared to the "sharp" global optimum.

On the other hand, fixing the parameter w_1 and w_2, we may also plot the labels predicted by the model given the samples. Here we plot the prediction from both the sharp minimum (Figure 1c) and the flat minimum (Figure 1d). The sharp minimum, even though it approximates the true label better, has some complex structures in its predicted labels, while the flat minimum seems to produce a simpler classification boundary.

While it is easy to make observations on toy examples, it is less straight-forward to make a quantitative statement when the model parameters and the number of training samples grow. In the following sections we try connect the local smoothness of the solution and model generalization capability. Section 3 briefly introduces some preliminaries on the learning theory. Section 4 talks about the assumptions and intuitions on how the model perturbation is related to the generalization as well as the Hessian of the solution. Section 5 dives into two specific types of perturbations: uniform and truncated Gaussian. Section 6 discusses the effect of re-parameterization on the proposed bound. Some empirical approximations and experiments are shown in Section 7 and 8.

3. Model Generalization Theory

We consider the general machine learning scenario. Suppose we have a labeled data set $\mathcal{S} = \{s_i = (x_i, y_i) \mid i \in \{1, \ldots, n\}, x_i \in \mathbb{R}^d, y_i \in \{0,1\}^k\}$, where (x_i, y_i) are sampled i.i.d. from a distribution $x_i, y_i \sim \mathfrak{D}_s$. We try to learn a function $f \in \mathfrak{F} : \mathcal{X} \to \mathcal{Y}$, such that the expected loss

$$L(f) = \mathbb{E}_{x,y\sim\mathfrak{D}_s} l(f, x, y)$$

is small, where $l : \mathfrak{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ is the loss function.

Since we do not know the distribution $\mathfrak{D}_s$, the expected loss $L(f)$ is hard to calculate directly. Instead usually the empirical loss

$$\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i)$$

is evaluated during the training procedure.

3.1 Rademacher Complexity

Minimizing the empirical loss

$$f^* = \arg\min_{f\in\mathfrak{F}} \hat{L}(f)$$

may lead to issues such as overfitting. In general, by the law of large number, for a fixed function $f \in \mathfrak{F}$, the empirical loss converges almost surely to the expected loss. However, when f is not fixed, i.e., f depends on the samples, and the number of samples is finite, classical learning theory suggests that the gap between the expected loss and the empirical

[1] the bound was approximated with $\eta = 39$ using inequality (13)

loss is bounded by the sum of the Rademacher complexity and a concentration tail (Shalev-Shwartz and Ben-David, 2014). The Rademacher complexity is defined as

$$\mathcal{R}_n(\mathfrak{F}, \mathfrak{D}_s) = \mathbb{E}_{x_i, y_i \sim \mathfrak{D}_s} \left[\mathbb{E}_\epsilon \sup_{f \in \mathfrak{F}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i l(f(x_i), y_i) \right],$$

where ϵ_is are i.i.d. Rademacher random variables.

Note the Rademacher complexity is only related to the function space $\mathfrak{F}$, the sample distribution $\mathfrak{D}_s$ and the number of samples n. This seems to suggest when the function class is very complex, the gap between the empirical loss and the expected loss will be large. Though the learning theory based on Rademacher complexity can explain the overfitting effect to some extent, for example, when the hypothesis space is overly complex, the generalization tends to be worse, it is not easy to explain some well-known empirical observations in today's deep learning experiments including:

- Over-parameterization.

  The hypothesis space of a deep learning network can easily get rich enough to represent any function on a finite sample set (Zhang et al., 2017). According to the bound based on the Rademacher complexity, the network may tend to overfit. However empirically those deep models generalize well.

- Different generalization behaviors for different local optima.

  The generalization bound based on Rademacher complexity holds uniformly for all hypothesis in the function class. On the other hand, it does not distinguish the generalization capabilities among different solutions. Obviously, there are "simple" solutions even if the whole function space is complex.

In this draft we will focus on the second empirical observations and give, to the best of our knowledge, a first explanation on behaviors of different local optima.

3.2 PAC-Bayes

Another line of theory discussing model generalization is PAC-Bayes (Mcallester, 2003) (McAllester, 1998) (McAllester, 1999) (Langford and Shawe-Taylor, 2002). The PAC-Bayes paradigm further assumes probability measures over the function class. In particular, it assumes a "posterior" distribution $\mathfrak{D}_f$ as well as a "prior" distribution π_f over the function class $\mathfrak{F}$. In this way the function is assumed to be sampled from a "posterior" distribution over $\mathfrak{F}$. As a consequence the expected loss is in terms of both the random draw of samples as well as the random draw of functions:

$$L(\mathfrak{D}_f, \mathfrak{D}_s) = \mathbb{E}_{f \sim \mathfrak{D}_f} \mathbb{E}_{x,y \sim \mathfrak{D}_s} l(f, x, y).$$

Correspondingly, the empirical loss in the PAC-Bayes paradigm is the expected loss over the draw of functions from the posterior:

$$\hat{L}(\mathcal{S}) = \mathbb{E}_{f \sim \mathfrak{D}_f} \frac{1}{n} \sum_{i=1}^{n} l(f, x_i, y_i).$$

PAC-Bayes theory suggests the gap between the expected loss and the empirical loss is bounded by a term that is related to the KL divergence between $\mathfrak{D}_f$ and π_f (McAllester, 1999) (Langford and Shawe-Taylor, 2002). In particular, if the function f is parameterized as $f(w)$ with $w \in \mathcal{W}$, when $\mathfrak{D}_w$ is perturbed around any w, we have the following PAC-Bayes bound (Seldin et al., 2012) (Seldin et al., 2011) (Neyshabur et al., 2017a) (Neyshabur et al., 2017b):

Theorem 1 (PAC-Bayes-Hoeffding Perturbation) *Let $l(f, x, y) \in [0, 1]$, and π be any fixed distribution over the parameters $\mathcal{W}$. For any $\delta > 0$ and $\eta > 0$, with probability at least $1 - \delta$ over the draw of n samples, for any w and any random perturbation u,*

$$\mathbb{E}_u[L(w+u)] \leq \mathbb{E}_u[\hat{L}(w+u)] + \frac{KL(w+u||\pi) + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \tag{1}$$

One may further optimize η to get a bound that scales approximately as $\mathbb{E}_u[L(w+u)] \lesssim \mathbb{E}_u[\hat{L}(w+u)] + 2\sqrt{\frac{KL(w+u||\pi)+\log\frac{1}{\delta}}{2n}}$ (Seldin et al., 2011). [2] A nice property of the perturbation bound (1) is it connects the generalization with the local properties around the solution w through some perturbation u around w. In particular, suppose $\hat{L}(w^*)$ is a local optima, when the perturbation level of u is small, $\mathbb{E}_u[\hat{L}(w^* + u)]$ tends to be small, but $KL(w^* + u|\pi)$ may be large since the posterior is too "focused" on a small neighboring area around w^*, and vice versa. As a consequence, we may need to search for an "optimal" perturbation level for u so that the bound is minimized.

4. Local Smoothness Assumptions

Keskar et al. (2016) investigate the local structures of the converged points for deep learning networks, and find that empirically the "sharpness" of the minima is closely related to the generalization property of the classifier. The sharp minimizers, which led to lack of generalization ability, are characterized by a significant number of large positive eigenvalues in $\nabla^2 f(x)$. In particular, they propose a local sharpness metric:

Definition 2 (Sharpness Metric) *(Keskar et al., 2016) Given $x \in \mathbb{R}^m$, $\epsilon > 0$ and $A \in \mathbb{R}^{m\times p}$, the (C_ϵ, A)-sharpness of f at x is defined as:*

$$\phi_{x,f}(\epsilon, A) := \frac{(\max_{y\in C_\epsilon} f(x+Ay)) - f(x)}{1 + f(x)} \times 100 \tag{2}$$

where $C_\epsilon = \{z \in \mathbb{R}^p : -\epsilon(|(A^+x)_i| + 1) \leq z_i \leq \epsilon(|(A^+x)_i| + 1), \quad \forall i \in \{1, 2, \ldots, p\}\}$, and A^+ is the pseudo inverse of A.

Other variants of the model generalization metrics are also proposed by Chaudhari et al. (2016) and Novak et al. (2018b).

Neyshabur et al. (2017a) suggests an "expected sharpness" based on the PAC-Bayes bound:

$$E_{u\sim N(0,\sigma^2)^m}[\hat{L}(w+u)] - \hat{L}(w) \tag{3}$$

[2] Since η cannot depend on the data, one has to build a grid and use the union bound.

They also point out the sharpness itself may not be enough to determine the generalization capability, but combining scales with sharpness one may get a control of the generalization. Similar connections are also found by Dziugaite and Roy (2017).

4.1 Smoothness Assumption over Hessian

While some researchers have discovered empirically the generalization ability of the models is related to the second order information around the local optima, to the best of our knowledge there is no work on how to connect the Hessian matrix $\nabla^2 \hat{L}(w)$ with the model generalization. In this section we introduce the assumption about the second-order smoothness, which is later used in our generalization bound.

Definition 3 (Hessian Lipschitz) *A twice differentiable function $f(\cdot)$ is ρ-Hessian Lipschitz if:*

$$\forall w_1, w_2, \|\nabla^2 f(w_1) - \nabla^2 f(w_2)\| \leq \rho \|w_1 - w_2\|, \tag{4}$$

where $\|\cdot\|$ is the operator norm.

The Hessian Lipschitz condition has been used in the numeric optimization community to model the second-order smoothness (Nesterov and Polyak, 2006) (Allen-Zhu and Orecchia, 2014). For the deep models it could be unrealistic to assume the Hessian Lipschitz condition holds for all $w \in \mathcal{W}$. Instead we make a local Hessian Lipschitz assumption:

Definition 4 (Local Hessian Lipschitz) *Function $\hat{L}(w)$ is ρ-Hessian Lipschitz in $Neigh_{\gamma,\epsilon}(w)$, where*

$$Neigh_{\gamma,\epsilon}(w) = \{v \mid |v_i - w_i| \leq \gamma |w_i| + \epsilon \quad \forall i\}$$

is a neighborhood around w defined by two positive constants γ and ϵ.

To simplify the notation in the draft we denote $\kappa_{\gamma,\epsilon}(w_i) = \gamma |w_i| + \epsilon$.

4.2 Connecting Generalization and Hessian

Suppose the empirical loss function $\hat{L}(w)$ satisfies the local Hessian Lipschitz condition, then by Lemma 1 in (Nesterov and Polyak, 2006), the perturbation of the function around a fixed point can be bounded by terms up to the third-order,

$$\hat{L}(w+u) \leq \hat{L}(w) + \nabla \hat{L}(w)^T u + \frac{1}{2} u^T \nabla^2 \hat{L}(w) u + \frac{1}{6}\rho \|u\|^3 \quad \forall u \;\; s.t. \;\; w+u \in Neigh_{\gamma,\epsilon}(w) \tag{5}$$

For perturbations with zero expectation, i.e., $\mathbb{E}[u] = 0$, the linear term in (5), $\mathbb{E}_u[\nabla^2 \hat{L}(w)^T u] = 0$. Because the perturbation u_i for different parameters are independent, the second order term can also be simplified.

$$\mathbb{E}_u\Big[\frac{1}{2} u^T \nabla^2 \hat{L}(w) u\Big] = \frac{1}{2} \sum_i \nabla_i^2 \hat{L}(w) \mathbb{E}[u_i^2], \tag{6}$$

where ∇_i^2 is simply the i-th diagonal element in Hessian. The following lemma is straightforward given (1),(5), and (6).

Lemma 5 *Suppose the loss function $l(f, x, y) \in [0, 1]$. Let π be any distribution on the parameters that is independent from the data. For any $\delta > 0$ and $\eta > 0$, with probability at least $1-\delta$ over the draw of n samples, for any w such that $\hat{L}(w)$ satisfies the local ρ-Hessian Lipschitz condition in $Neigh_{\gamma,\epsilon}(w)$, and any random perturbation u, s.t., $|u_i| \leq \kappa_{\gamma,\epsilon}(w_i)$ $\forall i$, $\mathbb{E}[u] = 0$, u_i and u_j are independent for any $i \neq j$, we have*

$$\mathbb{E}_u[L(w+u)] \leq \hat{L}(w) + \frac{1}{2}\sum_i \nabla_i^2 \hat{L}(w)\mathbb{E}[u_i^2] + \frac{\rho}{6}\mathbb{E}[\|u\|^3] + \frac{KL(w+u||\pi) + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \quad (7)$$

where ∇_i is the i-th diagonal element of $\nabla \hat{L}(w)$.

Note by extrema of the Rayleigh quotient, the quadratic term on the right hand side of inequality (5) is further bounded by

$$u^T \nabla^2 \hat{L}(w) u \leq \lambda_{max}(\nabla^2 \hat{L}(w))\|u\|^2. \quad (8)$$

This is consistent with the empirical observations of Keskar et al. (2016) that the generalization ability of the model is related to the eigenvalues of $\nabla^2 \hat{L}(w)$. The inequality (8) still holds even if the perturbations u_i and u_j are correlated. We add another lemma about correlated perturbations in Appendix (Lemma 9).

4.3 Tradeoff between Sharpness Metric and Generalization Power

If we look at the right hand side of the inequality (7), and compare it with (3) (Neyshabur et al., 2017a), we see

$$\mathbb{E}_u \hat{L}(w+u) - \hat{L}(w) \leq \mathcal{M}(w, \mathcal{D}_u) = \frac{1}{2}\sum_i \nabla_i^2 \hat{L}(w)\mathbb{E}[u_i^2] + \frac{\rho}{6}\mathbb{E}[\|u\|^3] \quad (9)$$

$\mathcal{M}(w, \mathcal{D}_u)$ can be interpreted as the sharpness metric of the empirical loss. It is closely related to the Hessian $\nabla^2 L(w)$, but it is also related to the perturbation distributions. Figure (2) shows when the perturbation is fixed how $\nabla^2 \hat{L}(w)$ can affect the term $\mathbb{E}_u \hat{L}(w)$.

The other term

$$\mathcal{G}_{\delta,n}(\eta, \mathcal{D}_{w+u}, \pi) = \frac{KL(w+u||\pi) + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \quad (10)$$

is related to the model generalization power in the original PAC-Bayes bound.

Ideally we would like both $\mathcal{M}(w, \mathcal{D}_u)$ and $\mathcal{G}_{\delta,n}(\eta, \mathcal{D}_{w+u}, \pi)$ to be small for better generalization capability. However, generally the perturbation distribution that leads to small $\mathcal{M}(w, \mathcal{D}_u)$ tends to have large $\mathcal{G}_{\delta,n}(\eta, \mathcal{D}_{w+u}, \pi)$ for a given prior. As we will see in the following sections, in the end we have to make trade-offs between the two terms.

5. Bounded Perturbations

Adding noise to the model for better generalization has proven successful both empirically and theoretically (Zhu et al., 2018) (Hoffer et al., 2017) (Jastrzębski et al., 2017) (Dziugaite and Roy, 2017) (Novak et al., 2018a). Instead of only minimizing the empirical loss, (Langford and Caruana, 2001) and (Dziugaite and Roy, 2017) assume different perturbation levels on different parameters, and minimize the generalization bound led by PAC-Bayes for better model generalization. However how to connect the noise distribution with the local optima structures, for example, $\nabla^2 L(w^*)$, and how that is related to the generalization power have not been examined.

Since the assumptions in Lemma (4.2) are local, the distributions of interest for the perturbation are necessarily bounded. In this section we investigate two special forms of perturbations, the uniform perturbation and truncated Gaussian, and provide closed-form scale estimation for the perturbation levels.

5.1 Uniform Distribution

Suppose $u_i \sim U(-\sigma_i, \sigma_i)$, and $\sigma_i \le \kappa_{\gamma,\epsilon}(w_i) \;\; \forall i$. That is, the "posterior" distribution of the model parameters are uniform distribution, and the distribution supports vary for different parameters. We also assume the perturbed parameters are bounded, i.e., $|w_i| + \kappa_{\gamma,\epsilon}(w_i) \le \tau_i \;\; \forall i$.[3] If we choose the priors π to be $u_i \sim U(-\tau_i, \tau_i)$, and then

$$KL(w+u||\pi) = \sum_i \log(\tau_i/\sigma_i) \tag{11}$$

Note $E[u_i^2] = \sigma_i^2/3$. Also we simplify the third order term in (7) by

$$\frac{\rho}{6}\mathbb{E}[\|u\|^3] \le \frac{\rho m^{1/2}}{6}\mathbb{E}[\|u\|_3^3] \le \frac{\rho m^{1/2}}{6}\sum_i \kappa_{\gamma,\epsilon}(w_i)\mathbb{E}[u_i^2] = \frac{\rho m^{1/2}}{18}\sum_i \kappa_{\gamma,\epsilon}(w_i)\sigma_i^2,$$

where we use the inequality $\|u\|_2 \le m^{\frac{1}{6}}\|u\|_3$ and m is the number of parameters. By Lemma (4.2), we get

$$\mathbb{E}_u[L(w+u)] \le \hat{L}(w) + \frac{1}{6}\sum_i \nabla_i^2 L(w)\sigma_i^2 + \frac{\rho m^{1/2}}{18}\sum_i \kappa_{\gamma,\epsilon}(w_i)\sigma_i^2 + \frac{\sum_i \log\frac{\tau_i}{\sigma_i} + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \tag{12}$$

If we assume $\hat{L}(w)$ is locally convex around w^* so that $\nabla_i^2 \hat{L}(w^*) \ge 0$ for all i. Solve for σ that minimizes the right hand side, and we have the following lemma:

Lemma 6 *Suppose the loss function $l(f,x,y) \in [0,1]$, and model weights are bounded $|w_i| + \kappa_{\gamma,\epsilon}(w_i) \le \tau_i \;\; \forall i$. For any $\delta > 0$ and η, with probability at least $1-\delta$ over the draw of n samples, for any $w^* \in \mathbb{R}^m$ such that $\hat{L}(w)$ is locally convex in $Neigh_{\gamma,\epsilon}(w^*)$ and $\hat{L}(w)$ satisfies the local ρ-Hessian Lipschitz condition in $Neigh_{\gamma,\epsilon}(w^*)$,*

$$\mathbb{E}_u[L(w^*+u)] \le \hat{L}(w^*) + \frac{m/2 + \sum_i \log\frac{\tau_i}{\sigma_i^*} + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \tag{13}$$

where $u_i \sim U(-\sigma_i^*, \sigma_i^*)$ are i.i.d. uniformly perturbed random variables, and

$$\sigma_i^*(w^*,\eta,\gamma) = \min\left(\sqrt{\frac{1}{\eta(\nabla_i^2 L(w^*)/3 + \rho m^{1/2}\kappa_{\gamma,\epsilon}(w_i^*)/9)}}, \kappa_{\gamma,\epsilon}(w_i^*)\right). \tag{14}$$

[3] One may also assume the same τ for all parameters for a simpler argument. The proof procedure goes through in a similar way.

In our experiment, we simply treat η as a hyper-parameter. Other other hand, one may further build a weighted grid over η and optimize for the best η (Seldin et al., 2011). In this way we reach the following theorem:

Theorem 7 *Under the conditions of Lemma 6, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of n samples, for any $w^* \in \mathbb{R}^m$ such that in $Neigh_{\gamma,\epsilon}(w^*)$, $\hat{L}(w)$ is locally convex and satisfies the local ρ-Hessian Lipschitz condition,*

$$\mathbb{E}_u[L(w^*+u)] \leq \hat{L}(w^*) + O\left(\sqrt{\frac{m + \sum_i \log \frac{\tau_i}{\sigma_i^*} + \log \frac{1}{\delta}}{n}}\right)$$

where $u_i \sim U(-\sigma_i^*, \sigma_i^*)$ are i.i.d. uniformly perturbed random variables, and

$$\sigma_i^*(w^*, \eta, \gamma) = \min\left(\sqrt{\frac{1}{\sqrt{mn}(\nabla_i^2 \hat{L}(w^*)/3 + \rho m^{1/2} \kappa_{\gamma,\epsilon}(w_i^*)/9)}}, \kappa_{\gamma,\epsilon}(w_i^*)\right) \tag{15}$$

Please see the appendix for the details of the proof.

5.2 Truncated Gaussian

Because the Gaussian distribution is not bounded but Lemma (4.2) requires bounded perturbation, we first truncate the distribution. The procedure of truncation is similar to the proof in (Neyshabur et al., 2017b) and (Mcallester, 2003).

Let $u \sim N(0, \Sigma)$, where Σ is a diagonal covariance matrix. Denote the truncated Gaussian as $N_{\gamma,\epsilon}(0, \Sigma)$. If $\tilde{u} \sim N_{\gamma,\epsilon}(0, \Sigma)$ then

$$\mathbb{P}_{\gamma,\epsilon}(\tilde{u}) = \frac{1}{Z}\begin{cases} p(u) & \text{if } \ |u_i| < \kappa_{\gamma,\epsilon}(w_i) \ \ \forall i \\ 0 & o.w. \end{cases} \tag{16}$$

If $\forall i \quad \sigma_i < \frac{\kappa_{\gamma,\epsilon}(w_i)}{\sqrt{2}\mathrm{erf}^{-1}(\frac{1}{2m})}$, by union bound $Z \geq 1/2$. Here erf^{-1} is the inverse Gaussian error function defined as $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$, and m is the number of parameters. Following a similar procedure as in the proof of Lemma 1 in (Neyshabur et al., 2017b),

$$KL(w+\tilde{u}||\pi) \leq 2(KL(w+u||\pi)+1) \tag{17}$$

Suppose the coefficients are bounded such that $\sum_i w_i^2 \leq \tau$, where τ is a constant. Choose the prior π as $N(0, \tau I)$, and we have

$$KL(w+u||\pi) \leq \frac{1}{2}(m \log \tau - \sum_i \log \sigma_i^2 - m + \frac{1}{\tau}\sum_i \sigma_i^2 + 1) \tag{18}$$

Notice that after the truncation the variance only becomes smaller, so the bound of (7) for the truncated Gaussian becomes

$$\begin{aligned}\mathbb{E}_u[L(w+\tilde{u})] \leq \hat{L}(w) &+ \frac{1}{2}\sum_i \nabla_i^2 L(w)\sigma_i^2 + \frac{\rho m^{1/2}}{6}\sum_i \kappa_{\gamma,\epsilon}(w_i)\sigma_i^2 \\ &+ \frac{m \log \tau - \sum_i \log \sigma_i^2 - m + \frac{1}{\tau}\sum_i \sigma_i^2 + 1 + 2\log\frac{1}{\delta}}{2\eta} + \frac{\eta}{2n}\end{aligned} \tag{19}$$

Again when $\hat{L}(w)$ is convex around w^* such that $\nabla^2 \hat{L}(w^*) \geq 0$, solve for the best σ_i and we get the following lemma:

Lemma 8 *Suppose the loss function $l(f, x, y) \in [0, 1]$, and model weights are bounded $\sum_i w_i^2 \leq \tau$. For any $\delta > 0$ and η, with probability at least $1 - \delta$ over the draw of n samples, for any $w^* \in \mathbb{R}^m$ such that in $Neigh_{\gamma,\epsilon}(w^*)$, $\hat{L}(w)$ is convex and satisfies the local ρ-Hessian Lipschitz condition,*

$$\mathbb{E}_u[L(w^* + \tilde{u})] \leq \hat{L}(w^*) + \frac{m \log \tau - \sum_i \log \sigma_i^2 + 1 + 2 \log \frac{1}{\delta}}{2\eta} + \frac{\eta}{2n} \tag{20}$$

where $\tilde{u} \sim N_{\gamma,\epsilon}(0, \Sigma^*)$ are random variables distributed as truncated Gaussian,

$$\sigma_i^* = \min\left(\sqrt{\frac{1}{\eta \nabla_i^2 \hat{L}(w^*) + \frac{\rho \eta m^{1/2}}{3} \kappa_{\gamma,\epsilon}(w_i^*) + \frac{1}{\tau}}}, \frac{\kappa_{\gamma,\epsilon}(w_i^*)}{\sqrt{2}\text{erf}^{-1}(\frac{1}{2m})}\right) \tag{21}$$

and σ_i^{*2} is the i-th diagonal element in Σ^*.

Again We have an extra term η, which may be further optimized over a grid to get a tighter bound. In our algorithm we treat η as a hyper-parameter instead.

6. On the Re-parameterization of RELU-MLP

Dinh et al. (2017) points out the spectrum of $\nabla^2 \hat{L}$ itself is not enough to determine the generalization power. One particular example is the multiple layer perceptron with RELU as the activations (RELU-MLP). For a two-layer RELU-MLP, denote w^1, and w^2 as the linear coefficients for the first and second layer. Clearly

$$\hat{L}(w^1, w^2) = \hat{L}(\alpha w^1, \alpha^{-1} w^2) \tag{22}$$

If cross entropy (negative log likelihood) is used as the loss function, under certain regularization conditions, if $p(x, y) = f(x, w^*)[y]$, i.e., w^* is the "true" parameter of the sample distribution, the change in Hessian to re-parameterization can be calculated as the outer product of the gradients, in this case

$$\nabla^2 \hat{L}(\alpha w^1, \alpha^{-1} w^2) = \begin{bmatrix} \alpha^{-1} I_{m^1} & 0 \\ 0 & \alpha I_{m^2} \end{bmatrix} \nabla^2 \hat{L}(w^1, w^2) \begin{bmatrix} \alpha^{-1} I_{m^1} & 0 \\ 0 & \alpha I_{m^2} \end{bmatrix} \tag{23}$$

In general our bound does not assume the loss function to be cross entropy loss. Also we do not assume the model is RELU-MLP. As a result we would not expect our bound stays exactly the same during the re-parameterization.

On the other hand, the optimal perturbation levels in our bound scales inversely during the scaling of parameters, so the bound only changes approximately with a speed of logarithmic factor. According to Lemma (6) and (8), if we use the optimal σ^* on the right hand side of the bound, $\nabla^2 \hat{L}(w)$, ρ, and w^* are all behind the logarithmic terms. As a consequence, for RELU-MLP, if we do the re-parameterization trick as in Dinh et al. (2017), the change of the bound is small.

Disclaim: Section 7 and 8 will be heuristic-based experiments and approximations. They are not rigorous.

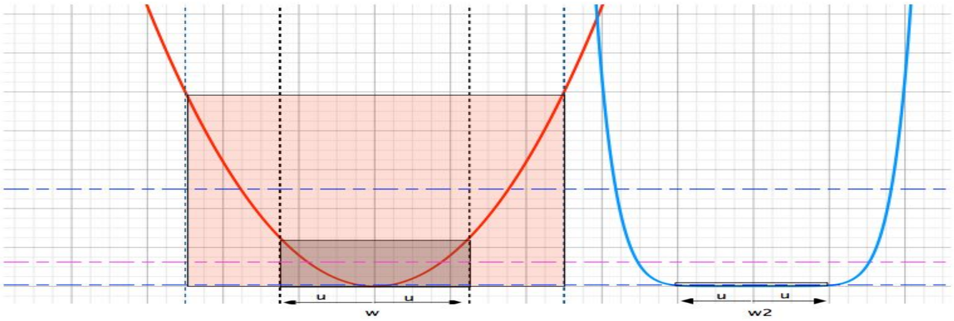
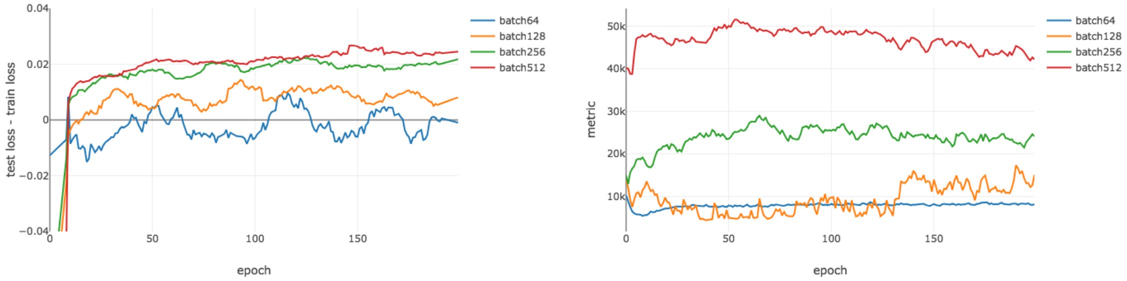


Figure 2: Sharpness Metric for $\hat{L}(w)$, 1-dimensional case. Fixing the perturbation level, larger $\nabla^2 \hat{L}(w)$ leads to larger $\mathcal{M}(w, \mathcal{D}_u)$.



(a) Test Loss - Train Loss (MNIST)

(b) $\Psi_{\gamma=0.1,\epsilon=0.1}$ (MNIST)

Figure 3: Generalization gap and $\Psi_{\gamma=0.1,\epsilon=0.1}$ as a function of epochs on MNIST for different batch sizes. SGD is used as the optimizer, and the learning rate is set as 0.1 for all configurations. As the batch size grows, $\Psi_{\gamma,\epsilon}(\hat{L}, w^*)$ gets larger. The trend is consistent with the true gap of losses.

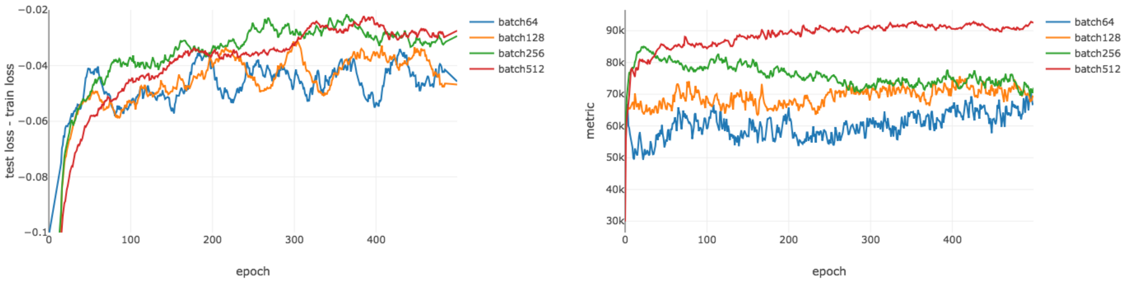
7. An Approximate Generalization Metric

Assuming $\hat{L}(w)$ is locally convex around w^*, so that $\nabla_i^2 \hat{L}(w^*) \geq 0$ for all i. If we look at Lemma 6, for fixed m and n, the only relevant term is $\sum_i \log \frac{\tau_i}{\sigma_i^*}$. Replacing the optimal σ^*, and using $\tau_i \sim |w_i| + \kappa_{\gamma,\epsilon}(w_i)$ to approximate τ_i, we come up with **PAC**-Bayes based **Gen**eralization metric, called pacGen,[4]

$$\Psi_{\gamma,\epsilon}(\hat{L}, w^*) = \sum_i \log\left((|w_i^*| + \kappa_{\gamma,\epsilon}(w_i^*)) \max\left(\sqrt{\nabla_i^2 \hat{L}(w^*) + \rho(w^*)\sqrt{m}\kappa_{\gamma,\epsilon}(w_i^*)}, \frac{1}{\kappa_{\gamma,\epsilon}(w_i^*)} \right) \right). \tag{24}$$

To calculate the metric on real-world data we need to estimate the diagonal elements of the Hessian $\nabla^2 \hat{L}$ as well as the Lipschitz constant ρ of the Hessian. For efficiency concern

[4] Even though we assume the local convexity in our metric, in application we may calculate the metric on every points. When $\nabla_i^2 \hat{L}(w^*) + \rho(w^*)\sqrt{m}\kappa_{\gamma,\epsilon}(w_i^*) < 0$ we simply treat it as 0.



(a) Test Loss - Train Loss (CIFAR-10)

(b) $\Psi_{\gamma=0.1,\epsilon=0.1}$ (CIFAR-10)

Figure 4: Generalization gap and $\Psi_{\gamma=0.1,\epsilon=0.1}$ as a function of epochs on CIFAR-10 for different batch sizes. SGD is used as the optimizer, and the learning rate is set as 0.01 for all configurations.

we follow Adam (Kingma and Ba, 2014) and approximate $\nabla_i^2 \hat{L}$ by $(\nabla \hat{L}[i])^2$. Also we use the exponential smoothing technique with $\beta = 0.999$ as in (Kingma and Ba, 2014).

To estimate ρ, we first estimate the Hessian of a randomly perturbed model $\nabla^2 \hat{L}(w+u)$[5], and then approximate ρ by $\rho = \max_i \frac{|\nabla_i^2 L(w+u_i) - \nabla_i^2 L(w)|}{|u_i|}$.

We used the same model without dropout from the PyTorch example [6]. We fix the learning rate as 0.1 and vary the batch size for training. The gap between the test loss and the training loss, and the metric $\Psi_{\gamma,\epsilon}(\hat{L}, w^*)$ are plotted in Figure 3. We had the same observation as in (Keskar et al., 2016) that as the batch size grows, the gap between the test loss and the training loss tends to get larger. Our proposed metric $\Psi_{\gamma,\epsilon}(\hat{L}, w^*)$ also shows the exact same trend. Note we do not use LR annealing heuristics as in (Goyal et al., 2017) which enables large batch training.

Similarly we also carry out experiment by fixing the training batch size as 256, and varying the learning rate. Figure 5 shows generalization gap and $\Psi_{\gamma,\epsilon}(\hat{L}, w^*)$ as a function of epochs. It is observed that as the learning rate decreases, the gap between the test loss and the training loss increases. And the proposed metric $\Psi_{\gamma,\epsilon}(\hat{L}, w^*)$ shows similar trend compared to the actual generalization gap.

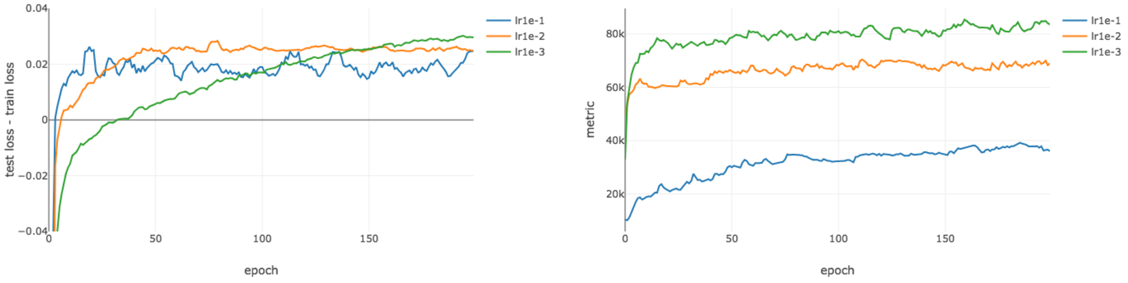
We also run the same model and experiment on CIFAR-10 (Krizhevsky et al.) just to demonstrate the effectiveness of the metric. We observed similar trends on CIFAR-10 as shown in Figure 4 and Figure 6.

8. A Perturbed Optimization Algorithm

The right hand side of (1) has $\mathbb{E}_u[\hat{L}(w+u)]$. This suggests rather than minimizing the empirical loss $\hat{L}(w)$, we should optimize the perturbed empirical loss $\mathbb{E}_u[\hat{L}(w+u)]$ instead for a better model generalization power. Adding perturbation to the model is not a new trick.

[5] In the experiment the gradients are taken w.r.t. w instead of $w+u$, and we ignore the difference between $\nabla_w^2 \hat{L}(w+u)$ and $\nabla_{w+u}^2 \hat{L}(w+u)$.

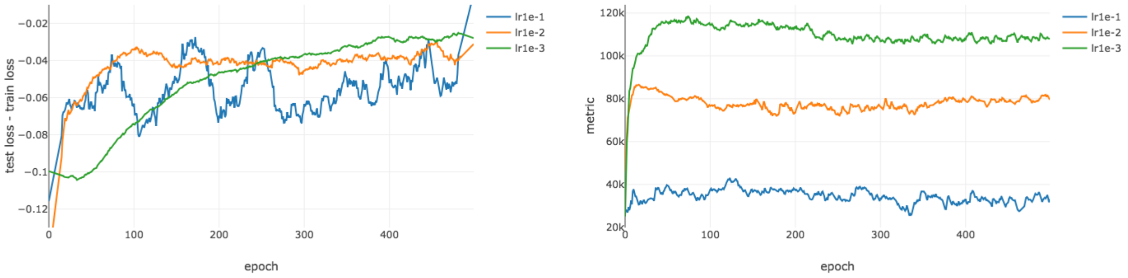
[6] `https://github.com/pytorch/examples/tree/master/mnist`



(a) Test Loss - Train Loss (MNIST)

(b) $\Psi_{\gamma=0.1,\epsilon=0.1}$ (MNIST)

Figure 5: Generalization gap and $\Psi_{\gamma=0.1,\epsilon=0.1}$ as a function of epochs on MNIST for different learning rates. SGD is used as the optimizer, and the batch size is set as 256 for all configurations. As the learning rate shrinks, $\Psi_{\gamma,\epsilon}(\hat{L}, w^*)$ gets larger. The trend is consistent with the true gap of losses.



(a) Test Loss - Train Loss (CIFAR-10)

(b) $\Psi_{\gamma=0.1,\epsilon=0.1}$ (CIFAR-10)

Figure 6: Generalization gap and $\Psi_{\gamma=0.1,\epsilon=0.1}$ as a function of epochs on CIFAR-10 for different learning rates. SGD is used as the optimizer, and the batch size is set as 256 for all configurations.

Most of the perturbation-based methods (Zhu et al., 2018) (Hoffer et al., 2017) (Jastrzębski et al., 2017) (Novak et al., 2018a) (Khan et al., 2018) are based on heuristic techniques and improvement in applications have already been observed empirically. Dziugaite and Roy (2017) first proposes to optimize for a better perturbation level from the PAC-Bayes bound, but their bound is not making use of the second order information. Also the best perturbation in (Dziugaite and Roy, 2017) is not close-form.

In this section we introduce a systematic way to perturb the model weights based on the PAC-Bayes bound. Again we use the same exponential smoothing technique as in Adam (Kingma and Ba, 2014) to estimate the Hessian $\nabla^2 \hat{L}$. To make the algorithm efficient, we ignore the third order part in the bound (7) so that we do not have to estimate the Lipschitz constant ρ of Hessian. The details of the algorithm is presented in (Algorithm 1), where we treat η as a hyper-parameter to be optimized using the validation set.

Algorithm 1 Perturbed OPT

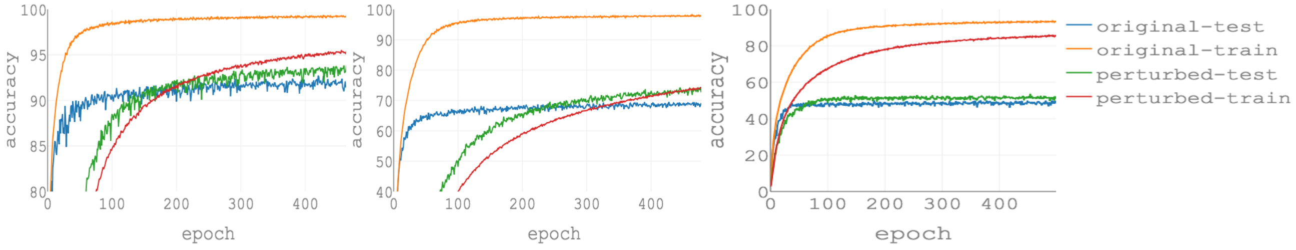
```
Require: $\eta$, $\gamma = 0.1$, $\beta_1 = 0.999$, $\beta_2 = 0.1$, $\epsilon$=1e-5.
Initialization: $\sigma_i \leftarrow 0$ for all $i$. $t \leftarrow 0$, $h_0 \leftarrow 0$
for epoch in $1, \ldots, N$ do
    for minibatch in one epoch do
        for all $i$ do
            if $t > 0$ then
                $\rho[i] \leftarrow \frac{|h_{t+1}[i] - h_t[i]|}{\|w_{t+1} - w_t\|}$
            $\kappa[i] \leftarrow \frac{\gamma}{\log(1+epoch)} |w_t[i]| + \epsilon$
            $\sigma_i \leftarrow \min\left(\frac{1}{\log(1+epoch)\sqrt{\eta(h_t[i] + \rho[i] \cdot \kappa[i])}}, \kappa[i]\right) \cdot \mathbf{1}_{|g_t[i]| < \beta_2}$
            $u_t[i] \sim U(-\sigma_i, \sigma_i)$(sample perturbation)
        $g_{t+1} \leftarrow \nabla_w \hat{L}_t(w_t + u_t)$ (get stochastic gradients w.r.t. perturbed loss)
        $h_{t+1} \leftarrow \beta_1 h_t + (1 - \beta_1) g_{t+1}^2$ (update second moment estimate)
        $w_{t+1} \leftarrow$ OPT$(w_t)$ (update $w$ using off-the-shell algorithms)
        $t \leftarrow t + 1$
```

Even though in theoretical analysis $E_u[\nabla \hat{L} \cdot u] = 0$, in applications, $\nabla \hat{L} \cdot u$ won't be zero especially when we only implement 1 trial of perturbation. On the other hand, if the gradient $\nabla \hat{L}$ is close to zero, then the first order term can be ignored. As a consequence, in (Algorithm 1) we only perturb the parameters that have small gradients whose absolute value is below β_2. For efficiency issues we used a per-parameter ρ_i capturing the variation of the diagonal element of Hessian. Also we decrease the perturbation level with a log factor as the epoch increases.

We compare the perturbed algorithm against the original optimization method on CIFAR-10, CIFAR-100 (Krizhevsky et al.), and Tiny ImageNet [7]. The results are shown in Figure 7. We use the Wide-ResNet (Zagoruyko and Komodakis, 2016) as the prediction model.[8] The depth of the chosen model is 58, and the widen-factor is set as 3. The dropout layers are turned off. For CIFAR-10 and CIFAR-100, we use Adam with a learning rate of

[7] `https://tiny-imagenet.herokuapp.com/`

[8] `https://github.com/meliketoy/wide-resnet.pytorch/blob/master/networks/wide_resnet.py`



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny ImageNet

Figure 7: Training and testing accuracy as a function of epochs on CIFAR-10, CIFAR-100 and Tiny ImageNet. For CIFAR, Adam is used as the optimizer, and the learning rate is set as 10^{-4}. For the Tiny ImageNet, SGD is used as the optimizer, and the learning rate is set as 10^{-2}.

10^{-4}, and the batch size is 128. For the perturbation parameters we use $\eta = 0.01$, $\gamma = 10$, and ϵ=1e-5. For Tiny ImageNet, we use SGD with learning rate 10^{-2}, and the batch size is 156. For the perturbed SGD we set $\eta = 100$, $\gamma = 1$, and ϵ=1e-5. Also we use the validation set as the test set for the Tiny ImageNet. We observe the the effect with perturbation appears similar to regularization. With the perturbation, the accuracy on the training set tends to decrease, but the test or the validation set increases.

9. Conclusion

We connect the smoothness of the solution with the model generalization in the PAC-Bayes framework. We prove that the generalization power of a model is related to the Hessian and the smoothness of the solution, the scales of the parameters, as well as the number of training samples. In particular, we prove that the best perturbation level scales roughly as $\frac{1}{\sqrt{\nabla^2 \hat{L}+\rho\sqrt{m}\kappa_{\gamma,\epsilon}(w_i)}}$, which mostly cancels out scaling effect in the re-parameterization suggested by (Dinh et al., 2017). To the best of our knowledge, this is the first work that integrate Hessian with the model generalization rigorously, and is also the first work explaining the effect of re-parameterization over the generalization rigorously. Based on our generalization bound, we propose a new metric to test the model generalization and a new perturbation algorithm that adjusts the perturbation levels according to the Hessian. Finally, we empirically demonstrate the effect of our algorithm is similar to a regularizer in its ability to attain better performance on unseen data.

10. Acknowledgement

The authors are grateful to Tengyu Ma, James Bradbury, Yingbo Zhou, and Bryan McCann for their helpful comments and suggestions on the manuscript.

References

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. pages 1–22, 2014. ISSN 2476-0757. doi: 10.23915/distill.00006. URL http://arxiv.org/abs/1407.1537.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *CoRR*, abs/1611.01838, 2016. URL http://arxiv.org/abs/1611.01838.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp Minima Can Generalize For Deep Nets. 2017. ISSN 1938-7228. URL http://arxiv.org/abs/1703.04933.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. 2017. URL http://arxiv.org/abs/1703.11008.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. URL http://arxiv.org/abs/1706.02677.

Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. URL http://arxiv.org/abs/1308.0850.

Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1064–1068, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR. URL http://proceedings.mlr.press/v65/harvey17a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014. URL http://arxiv.org/abs/1406.4729.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. 2017. ISSN 10495258. URL http://arxiv.org/abs/1705.08741.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three Factors Influencing Minima in SGD. pages 1–21, 2017. URL http://arxiv.org/abs/1711.04623.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. pages 1725–1732, 2014. doi: 10.1109/CVPR.2014.223. URL https://doi.org/10.1109/CVPR.2014.223.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016. URL http://arxiv.org/abs/1609.04836.

Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. pages 2616–2625, 2018. URL http://proceedings.mlr.press/v80/khan18a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012. URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

John Langford and Rich Caruana. (Not) Bounding the True Error. *Advances in Neural . . .*, 2001. ISSN 10495258. URL http://machinelearning.wustl.edu/mlpapers/paper{_}files/nips02-AA54.pdf.

John Langford and John Shawe-Taylor. Pac-bayes & margins. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 439–446, Cambridge, MA, USA, 2002. MIT Press. URL http://dl.acm.org/citation.cfm?id=2968618.2968674.

David Mcallester. Simplified pac-bayesian margin bounds. In *In COLT*, pages 203–215, 2003.

David A. McAllester. Some pac-bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 230–234, New York, NY, USA, 1998. ACM. ISBN 1-58113-057-0. doi: 10.1145/279943.279989. URL http://doi.acm.org/10.1145/279943.279989.

David A. McAllester. Pac-bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 164–170, New York, NY, USA, 1999. ACM. ISBN 1-58113-167-4. doi: 10.1145/307400.307435. URL http://doi.acm.org/10.1145/307400.307435.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. 2018. URL https://arxiv.org/abs/1806.08730. cite arxiv:1806.08730.

A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *Trans. Audio, Speech and Lang. Proc.*, 20(1):14–22, January 2012. ISSN 1558-7916. doi: 10.1109/TASL.2011.2109382. URL https://doi.org/10.1109/TASL.2011.2109382.

Yurii Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. *Math. Program.*, 108(1):177–205, August 2006. ISSN 0025-5610. doi: 10.1007/s10107-006-0706-8. URL https://doi.org/10.1007/s10107-006-0706-8.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring Generalization in Deep Learning. (Nips), 2017a. ISSN 10495258. URL http://arxiv.org/abs/1706.08947.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. (2017):1–9, 2017b. URL http://arxiv.org/abs/1707.09564.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and Generalization in Neural Networks: an Empirical Study. pages 1–21, 2018a. URL http://arxiv.org/abs/1802.08760.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=HJC2SzZCW.

Y. Seldin, F. Laviolette, and J. Shawe-Taylor. Pac-bayesian analysis of supervised, unsupervised, and reinforcement learning, 2012.

Yevgeny Seldin, François Laviolette, Nicolò Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. Pac-bayesian inequalities for martingales. *CoRR*, abs/1110.6886, 2011. URL http://arxiv.org/abs/1110.6886.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. pages 1631–1642, October 2013. URL http://www.aclweb.org/anthology/D13-1170.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2017. URL https://arxiv.org/abs/1611.03530.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from minima and regularization effects zhanxing. pages 1–15, 2018. URL http://arxiv.org/abs/1803.00195.

Appendix A. Proof of Lemma 6

Proof We rewrite the inequality (12) below

$$\mathbb{E}_u[L(w+u)] \leq \hat{L}(w) + \frac{1}{6}\sum_i \nabla_i^2 L(w)\sigma_i^2 + \frac{\rho m^{1/2}}{18}\sum_i(\gamma|w_i|+\epsilon)\sigma_i^2 + \frac{\sum_i \log\frac{\tau_i}{\sigma_i} + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \tag{25}$$

The terms related to σ_i on the right hand side of (25) are

$$\frac{1}{6}\nabla_i^2 L(w)\sigma_i^2 + \frac{\rho m^{1/2}}{18}(\gamma|w_i|+\epsilon)\sigma_i^2 - \frac{\log\sigma_i}{\eta} \tag{26}$$

Since the assumption is $\nabla_i^2\hat{L}(w^*) \geq 0$ for all i, $\nabla_i^2\hat{L}(w)+\rho m^{1/2}(\gamma|w_i|+\epsilon)/3 > 0$. Solving for σ that minimizes the right hand side of (25), and we have

$$\sigma_i^*(w,\eta,\gamma) = \min\left(\sqrt{\frac{1}{\eta(\nabla_i^2\hat{L}(w)/3+\rho m^{1/2}(\gamma|w_i|+\epsilon)/9)}}, \gamma|w_i|+\epsilon\right) \tag{27}$$

The term $\frac{1}{6}\sum_i \nabla_i^2 L(w)\sigma_i^2 + \frac{\rho m^{1/2}}{18}\sum_i(\gamma|w_i|+\epsilon)\sigma_i^2$ on the right hand side of (12) is monotonically increasing w.r.t. σ^2, so

$$\begin{aligned}&\frac{1}{6}\sum_i \nabla_i^2 L(w)\sigma_i^{*2} + \frac{\rho m^{1/2}}{18}\sum_i(\gamma|w_i|+\epsilon)\sigma_i^{*2}\\ &\leq \sum_i\left(\frac{1}{6}\nabla_i^2 L(w) + \frac{\rho m^{1/2}}{18}(\gamma|w_i|+\epsilon)\right)\frac{1}{\eta(\nabla_i^2\hat{L}(w)/3+\rho m^{1/2}(\gamma|w_i|+\epsilon)/9)}\\ &= \frac{m}{2\eta}\end{aligned} \tag{28}$$

Combine the inequality (28), and the equation (27) with (25), and we complete the proof.

■

Appendix B. Proof of Theorem 7

Proof

Combining (15) and (12), we get

$$\mathbb{E}_u[L(w+u)] \leq \hat{L}(w) + \frac{1}{2}\sqrt{\frac{m}{n}} + \frac{\sum_i \log\frac{\tau_i}{\sigma_i^*} + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n}$$

The following proof is similar to the proof of Theorem 6 in (Seldin et al., 2011). Note the η in Lemma (6) cannot depend on the data. In order to optimize η we need to build a grid of the form

$$\eta_j = e^j\sqrt{2n\log\frac{1}{\delta_j}}$$

for $j \geq 0$.

For a given value of $\sum_i \log \frac{\tau_i}{\sigma_i^*}$, we pick η_j, such that

$$j = \left\lfloor \frac{1}{2} \log \left(\frac{\sum_i \log \frac{\tau_i}{\sigma_i^*}}{\log \frac{1}{\delta_j}} + 1 \right) \right\rfloor$$

where $\lfloor x \rfloor$ is the largest integer value smaller than x. Set $\delta_j = \delta 2^{-(j+1)}$, and take a weighted union bound over η_j-s with weights $2^{-(j+1)}$, and we have with probability at least $1 - \delta$,

$$\mathbb{E}_u[L(w+u)] \leq \hat{L}(w) + \frac{1}{2}\sqrt{\frac{m}{n}} + (1+1/e)\sqrt{\frac{\sum_i \log \frac{\tau_i}{\sigma_i^*} + \log \frac{1}{\delta} + \frac{\log 2}{2}\left(2 + \log\left(\frac{\sum_i \log \frac{\tau_i}{\sigma_i^*}}{\log \frac{1}{\delta}} + 1\right)\right)}{2n}}$$

Simplify the right hand side and we complete the proof. ■

Appendix C. Proof of Lemma 8

Proof

We first rewrite the inequality (19) below:

$$\mathbb{E}_u[L(w+\tilde{u})] \leq \hat{L}(w) + \frac{1}{2}\sum_i \nabla_i^2 L(w)\sigma_i^2 + \frac{\rho m^{1/2}}{6}\sum_i (\gamma|w_i| + \epsilon)\sigma_i^2 + \frac{m \log \tau - \sum_i \log \sigma_i^2 - m + \frac{1}{\tau}\sum_i \sigma_i^2 + 1 + 2\log\frac{1}{\delta}}{2\eta} + \frac{\eta}{2n}$$

The terms related to σ_i on the right hand side of (19) is

$$\left(\frac{1}{2}\nabla_i^2 L(w) + \frac{\rho m^{1/2}}{6}(\gamma|w_i| + \epsilon) + \frac{1}{2\tau\eta}\right)\sigma_i^2 - \frac{\log \sigma_i^2}{2\eta} \tag{29}$$

Take gradients w.r.t. σ_i, when $\nabla_i^2 \hat{L} \geq 0$, we get the optimal σ_i^*,

$$\sigma_i^* = \min\left(\sqrt{\frac{1}{\eta \nabla_i^2 \hat{L}(w^*) + \frac{\rho \eta m^{1/2}}{3}(\gamma|w_i^*| + \epsilon) + \frac{1}{\tau}}}, \frac{\gamma|w_i^*| + \epsilon}{\sqrt{2}\mathrm{erf}^{-1}(\frac{1}{2m})}\right)$$

Note the first term in (29) is monotonously increasing w.r.t. σ_i, so

$$\begin{aligned}&\left(\frac{1}{2}\nabla_i^2 L(w) + \frac{\rho m^{1/2}}{6}(\gamma|w_i| + \epsilon) + \frac{1}{2\tau\eta}\right)\sigma_i^{*2} \\ &\leq \left(\frac{1}{2}\nabla_i^2 L(w) + \frac{\rho m^{1/2}}{6}(\gamma|w_i| + \epsilon) + \frac{1}{2\tau\eta}\right)\frac{1}{\eta \nabla_i^2 \hat{L}(w^*) + \frac{\rho \eta m^{1/2}}{3}(\gamma|w_i^*| + \epsilon) + \frac{1}{\tau}} \\ &= \frac{1}{2\eta}\end{aligned} \tag{30}$$

Summing over m parameters and combine (19), we complete the proof.

■

Appendix D. A Lemma about Eigenvalues of Hessian and Generalization

Lemma 9 *Suppose the loss function $l(f, x, y) \in [0, 1]$. Let π be any distribution on the parameters that is independent from the data. For any $\delta > 0$ and $\eta > 0$, with probability at least $1 - \delta$ over the draw of n samples, for any local optimal w^* such that $\nabla \hat{L}(w^*) = 0$, $\hat{L}(w)$ satisfies the local ρ-Hessian Lipschitz condition in $Neigh_{\gamma,\epsilon}(w^*)$, and any random perturbation u, s.t., $|u_i| \leq \gamma |w_i^*| + \epsilon \ \ \forall i$, we have*

$$\mathbb{E}_u[L(w^* + u)] \leq \hat{L}(w^*) + \frac{1}{2}\lambda_{max}\left(\nabla^2 \hat{L}(w^*)\right)\sum_i \mathbb{E}[u_i^2] + \frac{\rho}{6}\mathbb{E}[\|u\|^3] \\ + \frac{KL(w^* + u||\pi) + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n}. \tag{31}$$

Proof The proof of the Lemma 9 is straight-forward. Since $\nabla \hat{L}(w^*) = 0$, the first order term is zero at the local optimal point even if $\mathbb{E}[u] \neq 0$. By extrema of the Rayleigh quotient, the quadratic term on the right hand side of inequality (5) is further bounded by

$$u^T \nabla^2 \hat{L}(w) u \leq \lambda_{max}\left(\nabla^2 \hat{L}(w)\right)\|u\|^2. \tag{32}$$

Due to the linearity of the expected value,

$$\mathbb{E}[u^T \nabla^2 \hat{L}(w) u] \leq \lambda_{max}\left(\nabla^2 \hat{L}(w)\right)\sum_i \mathbb{E}[u_i^2], \tag{33}$$

which does not assume independence among the perturbations u_i and u_j for $i \neq j$.

■