Assessing Local Generalization Capability in Deep Models

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

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 a model, as well as an algorithm that optimizes the perturbed model accordingly.

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). Despite often having many more parameters than number of training samples, deep models generalize well (Huang et al., 2017).

Classical learning theory suggests that model generalization capability should be closely related to the "complexity" of the hypothesis space, usually measured in terms of number of parameters, Rademacher complexity or VC-dimension. However, this theoretical result contradicts the empirical observation that over-parameterized models generalize well on unseen test data. For example, over-parameterized neural networks can fit any function of sample size n, making the Rademacher complexity large, but empirically those neural networks generalize well (Zhang et al., 2016). Indeed, even if the hypothesis space is complex, the final solution learned from a given training set may still be simple. This suggests the generalization capability of the model is also related to the property of the solution.

Keskar et al. (2017) and Chaudhari et al. (2017) 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 w^*, and that large eigenvalues of the $\nabla^2 L(w^*)$ often leads to poor model generalization. Also, (Keskar et al., 2017), (Chaudhari et al., 2017) and (Novak et al., 2018b) introduce several different metrics to measure the "sharpness" of the solution and demonstrate the connection between these sharpness metrics and the generalization empirically. Dinh et al. (2017) later point 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 Bayesian analysis. Mackay (1995) first introduced Taylor expansion to approximate the (log) posterior of the parameters given the data, and considered the second-order term, characterized by the Hessian of the loss function, as a way of evaluating the model simplicity, or "Occam factor". Recently Smith and Le (2018) use this factor to penalize sharp minima, and determine the optimal batch size. Germain et al. (2016) connect the PAC-Bayes bound and the Bayesian marginal likelihood when the loss is (bounded) negative log-likelihood, which leads to an alternative perspective on Occam's razor. (Langford and Caruana, 2001), and more recently, (Harvey et al., 2017) (Neyshabur et al., 2017) (Neyshabur et al., 2018) use PAC-Bayes bound to analyze the generalization behavior of the deep models.

Since the PAC-Bayes bound holds uniformly for all "posteriors", it also holds for some particular "poste-

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

rior" (i.e. the solution parameter perturbed with noise). This provides a natural way to incorporate the local properties of the solution into the generalization analysis. In particular, Neyshabur et al. (2017) suggest to use the difference between the perturbed loss and the empirical loss as the sharpness metric. Dziugaite and Roy (2017) try to optimize the PAC-Bayes bound instead for a better model generalization. Still some fundamental questions remain unanswered.

In this paper we are interested in the following question:

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

The question above is also related to several practical questions, for example, during training how to balance the sharpness of the local optima and the empirical loss? It has been observed that adding perturbation in training can help boosting the generalization performance (Zhu et al., 2018) (Jastrzębski et al., 2017), but how to choose the perturbation level for different parameters remains unknown. We try to answer these questions from the PAC-Bayes perspective. Under mild assumptions on the Hessian of the loss function, we prove that the generalization error of the model is related to this Hessian, the Lipschitz constant of the Hessian, the scales of the parameters, and the number of training samples. Our analysis also gives rise to a new metric for generalization, which we show can be use to identify an approximately optimal perturbation level to aid generalization. Interestingly, the latter 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. We construct a small 2-dimensional sample set from a mixture of 3 Gaussians, and then binarize the labels by thresholding them from the median value. The sample distribution is shown in Figure 1b. For the model we use a 5-layer MLP with sigmoid as the activation and cross entropy as the loss. There are no bias terms in the linear layers, and the weights are shared. For the shared 2-by-2 linear coefficient matrix, we treat two entries as constants and optimize the other 2 entries. In this way the whole model has only 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 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.

3 PAC-Bayes and Model Generalization

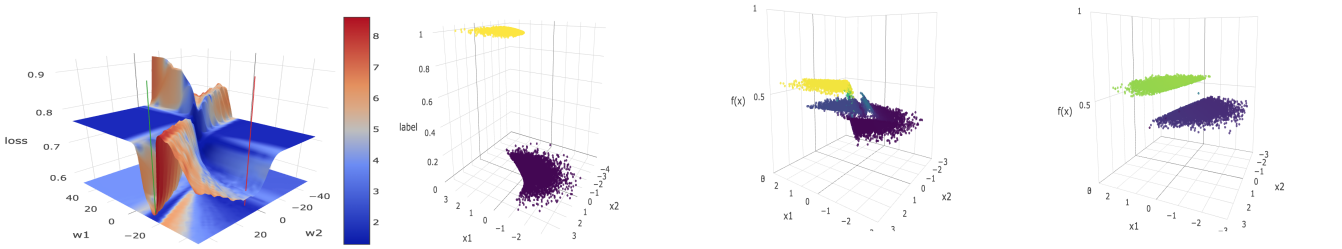
We consider the supervised learning in PAC-Bayes scenario (McAllester, 2003) (McAllester, 1998) (McAllester, 1999) (Langford and Shawe-Taylor, 2002). Suppose we have a labeled data set $\mathcal{S} = \{s_i = (x_i, y_i) \mid i \in \{1, \dots, n\}, x_i \in \mathbb{R}^d, y_i \in \{0,1\}^k\}$, where (x_i, y_i) are sampled i.i.d. from a distribution $x_i, y_i \sim \mathfrak{D}_s$.

The PAC-Bayes paradigm assumes probability measures over the function class $\mathfrak{F} : \mathcal{X} \to \mathcal{Y}$. In particular, it assumes a "posterior" distribution $\mathfrak{D}_f$ as well as a "prior" distribution π_f over the function class $\mathfrak{F}$. We are interested in minimizing the expected loss, in terms of both the random draw of samples as well as the random draw of functions:

$$L(\mathfrak{D}_f, \mathfrak{D}_s) = \mathbb{E}_{f \sim \mathfrak{D}_f} \mathbb{E}_{x,y \sim \mathfrak{D}_s} l(f, x, y). \tag{1}$$

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

[1] the bound was approximated with $\eta = 39$ using inequality (11)



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

from the posterior:

$$\hat{L}(\mathfrak{D}_f, \mathcal{S}) = \mathbb{E}_{f\sim\mathfrak{D}_f} \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i) \tag{2}$$

PAC-Bayes theory suggests the gap between the expected loss (1) and the empirical loss (2) 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., 2012b) (Seldin et al., 2012a) (Neyshabur et al., 2017) (Neyshabur et al., 2018):

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,*

$$\mathbb{E}_u[L(w+u)] \leq \mathbb{E}_u[\hat{L}(w+u)] + \frac{KL(w+u||\pi) + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \tag{3}$$

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., 2012a).[2] A nice property of the perturbation bound (3) 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 optimum, 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.

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

4 Main Result

While some researchers have already 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 rigorously. In this section we introduce the local smoothness assumption, as well as our main theorem.

It may be unrealistic to assume global smoothness properties for the deep models. Usually the assumptions only hold in a small local neighborhood $Neigh(w^*)$ around a reference point w^*. In this paper we define the neighborhood set as $Neigh_\kappa(w^*) = \{w \mid |w_i - w_i^*| \leq \kappa_i \ \ \forall i\}$, where $\kappa_i \in \mathbb{R}^+$ is the "radius" of the i-th coordinate. In our draft we focus on a particular type of radius $\kappa_i(w^*) = \gamma|w_i^*| + \epsilon$, but our argument holds for other types of radius, too.

In order to get a control of the deviation of the optimal solution we need to assume in $Neigh_{\gamma,\epsilon}(w^*)$, the empirical loss function $\hat{L}$ in (2) is Hessian Lipschitz, which is defined as:

Definition 1 (Hessian Lipschitz)**.** *A twice differentiable function $f(\cdot)$ is ρ-Hessian Lipschitz if:*

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

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

The Hessian Lipschitz condition has been used in the numeric optimization community to model the smoothness of the second-order gradients (Nesterov and Polyak, 2006) (Carmon et al., 2018) (Jin et al., 2018). In the rest of the draft we always assume the following:

Assumption 1. *In $Neigh_\kappa(w^*)$ the empirical loss $\hat{L}(w)$ defined in (2) is convex, and ρ-Hessian Lipschitz.*

For the uniform perturbation, the following theorem holds:

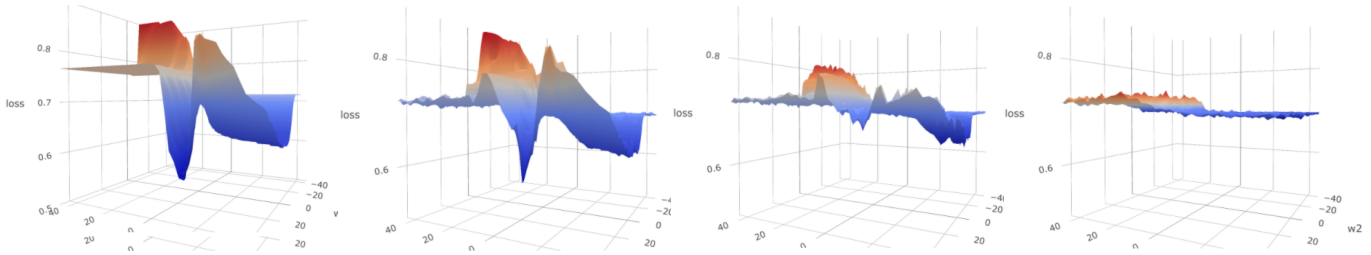


Figure 2: Loss Landscape as the perturbation level increases. From left to right: $\eta = 1000$, 0.1, 0.01, 1e-5.

Theorem 2. *Suppose the loss function $l(f, x, y) \in [0, 1]$, and model weights are bounded $|w_i| + \kappa_i(w) \leq \tau_i \quad \forall i$. With probability at least $1 - \delta$ over the draw of n samples, for any $\check{w} \in \mathbb{R}^m$ such that assumption 1 holds*

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

where $u_i \sim U(-\check{\sigma}_i, \check{\sigma}_i)$ are i.i.d. uniformly distributed random variables, and $\check{\sigma}_i(\check{w}, \eta, \gamma) =$

$$\min\left(\sqrt{\frac{1}{\sqrt{mn}(\nabla^2_{i,i}\hat{L}(\check{w})/3 + \rho m^{1/2}\kappa_i(\check{w})/9)}}, \kappa_i(\check{w})\right) \tag{4}$$

Theorem 2 says if we choose the perturbation levels carefully, the expected loss of a uniformly perturbed model is controlled. The bound is related to the diagonal element of Hessian (logarithmic), the Lipschitz constant ρ of the Hessian (logarithmic), the neighborhood scales characterized by κ (logarithmic), the number of parameters m, and the number of samples n. Also roughly the perturbation level is inversely related to $\sqrt{\nabla^2_{i,i}\hat{L}}$, suggesting the model be perturbed more along the coordinates that are "flat".[3]

Similar argument can be made on the truncated Gaussian perturbation, which is presented in Appendix C. In the next section we walk through some intuitions of our arguments.

5 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,

$$\hat{L}(w + u) \leq \hat{L}(w) + \nabla\hat{L}(w)^T u + \frac{1}{2}u^T\nabla^2\hat{L}(w)u + \frac{1}{6}\rho\|u\|^3 \quad \text{for} \quad w + u \in Neigh_\kappa(w) \tag{5}$$

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

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

where $\nabla^2_{i,i}$ is simply the i-th diagonal element in Hessian.

Considering (3),(5) and assumption 1, it is straightforward to see the bound below holds with probability at least $1 - \delta$

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

Suppose $u_i \sim U(-\sigma_i, \sigma_i)$, and $\sigma_i \leq \kappa_i(w) \quad \forall i$. That is, the "posterior" distributions 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., $|w_i| + \kappa_i(w) \leq \tau_i \quad \forall i$.[4] If we choose the prior π to be $u_i \sim U(-\tau_i, \tau_i)$, and then $KL(w + u\|\pi) = \sum_i \log(\tau_i/\sigma_i)$.

When γ is small, the third order terms $\frac{\rho}{6}\mathbb{E}[\|u\|^3]$ are small compared to the second order terms $\frac{1}{2}\sum_i \nabla^2_i\hat{L}(w)\mathbb{E}[u_i^2]$. In this case we bound the third

[3] Unfortunately the bound in theorem 2 does not explain the over-parameterization phenomenon since when $m \gg n$ the right hand side explodes.

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

order terms simply by

$$\frac{\rho}{6}\mathbb{E}[\|u\|^3] \le \frac{\rho m^{1/2}}{6}\mathbb{E}[\|u\|_3^3]$$
$$\le \frac{\rho m^{1/2}}{6}\sum_i \kappa_i(w)\mathbb{E}[u_i^2] = \frac{\rho m^{1/2}}{18}\sum_i \kappa_i(w)\sigma_i^2, \quad (8)$$

where we use the inequality $\|u\|_2 \le m^{\frac{1}{6}}\|u\|_3$ and m is the number of parameters. Pluging in (7), we get

$$\mathbb{E}_u[L(w+u)] \le \hat{L}(w) + \frac{1}{6}\sum_i \nabla^2_{i,i}L(w)\sigma_i^2$$
$$+ \frac{\rho m^{1/2}}{18}\sum_i \kappa_i(w)\sigma_i^2 + \frac{\sum_i \log\frac{\tau_i}{\sigma_i} + \log\frac{1}{\delta}}{\eta} + \frac{\eta}{2n} \quad (9)$$

If $\nabla^2_{i,i}\hat{L}(w) + \rho m^{1/2}(\gamma|w_i| + \epsilon)/3 > 0$, solve for σ that minimizes the right hand side, and we have

$$\sigma_i^*(w,\eta,\gamma) =$$
$$\min\left(\sqrt{\frac{1}{\eta(\nabla^2_{i,i}L(w)/3 + \rho m^{1/2}(\gamma|w_i| + \epsilon)/9)}}, \gamma|w_i| + \epsilon\right) \quad (10)$$

Otherwise if $\nabla^2_{i,i}L(w) + \rho m^{1/2}(\gamma|w_i| + \epsilon)/3 \le 0$, $\sigma_i^*(w,\eta,\gamma) = \gamma|w_i| + \epsilon$.

Equation (10) suggests that the optimal perturbation level approximately decreases with a speed of $1/\sqrt{\nabla^2_{i,i}\hat{L}(w)}$ as the corresponding Hessian diagonal increases.

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

Lemma 3. *Suppose the loss function $l(f,x,y) \in [0,1]$, and model weights are bounded $|w_i| + \kappa_i(w) \le \tau_i \quad \forall i$. Given any $\delta > 0$ and $\eta > 0$, with probability at least $1-\delta$ over the draw of n samples, for any $w^* \in \mathbb{R}^m$ such that assumption 1 holds,*

$$\mathbb{E}_u[L(w^*+u)] \le \hat{L}(w^*) + \frac{m/2 + \sum_i \log\frac{\tau_i}{\sigma_i^*} + \log\frac{1}{\delta}}{\eta}$$
$$+ \frac{\eta}{2n} \quad (11)$$

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

$$\min\left(\sqrt{\frac{1}{\eta(\nabla^2_{i,i}L(w^*)/3 + \rho m^{1/2}\kappa_i(w^*)/9))}}, \kappa_i(w^*)\right). \quad (12)$$

Figure (2) shows the effect of increasing the perturbation levels, i.e., decreasing η, on the toy example. It is observed that as the perturbation level increases, the "flat" minimum becomes lower compared to the "sharp" minimum. However if the perturbation level is too high the whole loss surface will be smoothed out. Thus a reasonable level of perturbation is desired. In our experiment, we simply treat η as a hyper-parameter.

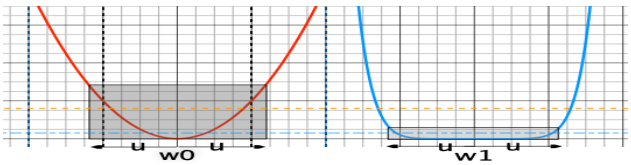


Figure 3: Sharpness Metric for $\hat{L}(w)$, 1-dimensional case. If we fix the perturbation level and integrate over the window of u around w, the flat minimum has a lower loss value, so sharper local minimum leads to larger $\mathcal{M}(\mathcal{S}, w, \mathfrak{D}_u)$.

On other hand, one may further build a weighted grid over η and optimize for the best η (Seldin et al., 2012a). That leads to Theorem 2. Details of the proof are presented in the Appendix D and E.

5.1 Generalization and Spectrum of Hessian

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

$$u^T\nabla^2\hat{L}(w)u \le \lambda_{max}(\nabla^2\hat{L}(w))\|u\|^2. \quad (13)$$

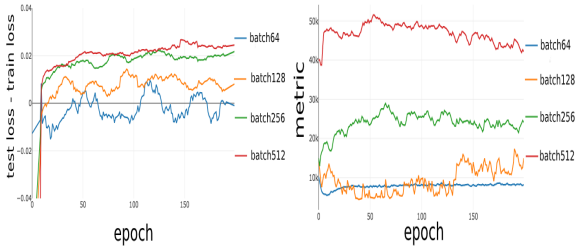
This is consistent with Keskar et al. (2017) 's empirical observations that the generalization ability of the model is related to the eigenvalues of $\nabla^2\hat{L}(w)$. The inequality (13) still holds even if the perturbations u_i and u_j are correlated. This suggests the following lemma:

Lemma 4. *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. Given $\delta > 0$ $\eta > 0$, with probability at least $1-\delta$ over the draw of n samples, for any local optimal w^* such that assumption 1 holds, we have*

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

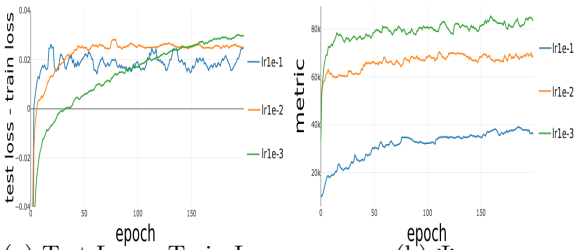
where u is any bounded perturbation s.t. $w^*+u \in Neigh_\kappa(w^*)$.

Note Lemma 4 does not make any zero-centering or independence assumption about the perturbations. The detailed proof is in Appendix H.



(a) Test Loss - Train Loss (b) Ψ_κ

Figure 4: Generalization gap and Ψ_κ as a function of epochs for different batch sizes on MNIST. SGD is used as the optimizer. The learning rate is set as 0.1 for all configurations.



(a) Test Loss - Train Loss (b) Ψ_κ

Figure 5: Generalization gap and Ψ_κ as a function of epochs for different learning rates on MNIST. SGD is used as the optimizer. The batch size is set as 256 for all configurations.

5.2 Tradeoff between Sharpness and Divergence

If we look at the right hand side of the inequality (7), and compare it with the inequality (5) in (Neyshabur et al., 2017), we see

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

$\mathcal{M}(\mathcal{S}, w, \mathfrak{D}_u)$ can be interpreted as the "sharpness" of the empirical loss. It is closely related to the Hessian $\nabla^2\hat{L}(w)$, but it is also related to the perturbation distributions $\mathfrak{D}_u$. Figure 3 visualizes the $\mathcal{M}(\mathcal{S}, w, \mathfrak{D}_u)$ of "flat" and "sharp" minima. The other term

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

is related to the divergence between the posterior and prior distributions of the parameters.

Ideally we would like both $\mathcal{M}(\mathcal{S}, w, \mathfrak{D}_u)$ and $\mathcal{G}_{\delta,n}(\eta, \mathfrak{D}_{w+u}, \pi)$ to be small for better generalization capability. However, generally the perturbation distribution that leads to small $\mathcal{M}(\mathcal{S}, w, \mathfrak{D}_u)$ tends to have large $\mathcal{G}_{\delta,n}(\eta, \mathfrak{D}_{w+u}, \pi)$ 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 by choosing the right level of perturbations for each parameter.

6 Comparison to Previous Works

Generalization of deep models has been investigated recently from different perspectives. Bartlett et al. (2017) bound the generalization gap by product of spectral norms of the coefficients based on techniques related to Rademacher complexity. Similarly Neyshabur et al. (2018) derive a spectral norm bound from the PAC-Bayes framework. Both works get to a frequentist bound on the generalization gap, where the expectation in the population loss is only over the random draw of samples. However the loss in our bound is an expectation over both "posterior" of the function as well as the sample as defined in (1).

Even though the worst-case solution of the neural network function class could be bad, in application due to some implicit bias the optimizer may not return the worst-case solution (Soudry et al., 2018) (Arora et al., 2019) (Gunasekar et al., 2017). According to our bound as long as the solution is locally smooth the model can generalize. Our bound is not directly comparable to the previous works since most previous works do not depend on local Hessian.

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. In particular, for a multi-layer perceptron with RELU as the activation function, one may re-parameterize the model and scale the Hessian spectrum arbitrarily without affecting the model prediction and generalization when cross entropy (negative log likelihood) is used as the loss and w^* is the "true" parameter of the sample distribution.

In general our bound does not assume the loss to be the cross entropy. 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 when the parameters scale, so the bound only changes approximately with a speed of logarithmic factor. According to Lemma (3), 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 function. As a consequence, for RELU-MLP, if we do the re-parameterization trick, the change of the bound is small.

7 An Approximate Generalization Metric

Assuming $\hat{L}(w)$ is locally convex around w^*, so that $\nabla^2_{i,i}\hat{L}(w^*) \geq 0$ for all i. If we look at Lemma 3, for fixed m and n, the only relevant term is $\sum_i \log \frac{\tau_i}{\sigma_i^*}$. Replacing the optimal σ^*, and using $|w_i| + \kappa_i(w)$ to approximate τ_i, we come up with **PAC**-Bayes based **Gen**eralization metric, called pacGen.[5]

$$\Psi_\kappa(\hat{L}, w^*) = \sum_i \log\left(|w_i^*| + \kappa_i(w^*)\right) \cdot \max\left(\sqrt{\nabla^2_{i,i}\hat{L}(w^*) + \rho(w^*)\sqrt{m}\kappa_i(w^*)}, \frac{1}{\kappa_i(w^*)}\right).$$

A self-explained toy example is displayed in Figure 1. 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 we follow Adam (Kingma and Ba, 2014) and approximate $\nabla^2_{i,i}\hat{L}$ 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 $\nabla^2\hat{L}(w+u)$, and then approximate ρ by $\rho = \max_i \frac{|\nabla_i^2 L(w+u_i) - \nabla_i^2 L(w)|}{|u_i|}$. For the neighborhood radius κ we use $\gamma = 0.1$ and $\epsilon = 0.1$ for all the experiments in this section.

We used the same model without dropout from the PyTorch MNIST example [6]. Fixing the learning rate as 0.1, we vary the batch size for training. The gap between the test loss and the training loss, and the metric $\Psi_\kappa(\hat{L}, w^*)$ are plotted in Figure 4. We had the same observation as in (Keskar et al., 2017) that as the batch size grows, the gap between the test loss and the training loss tends to get larger. Our proposed metric $\Psi_\kappa(\hat{L}, w^*)$ 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 $\Psi_\kappa(\hat{L}, w^*)$ 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 $\Psi_\kappa(\hat{L}, w^*)$ shows similar trend compared to the actual generalization gap.

[5] Even though we assume the local convexity in our metric, in application we may calculate the metric on every points. When $\nabla^2_{i,i}\hat{L}(w^*) + \rho(w^*)\sqrt{m}\kappa_i(w^*) < 0$ we simply treat it as 0.

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

8 A Perturbed Optimization Algorithm

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 integrate the smoothness property of the local optima is not clear.

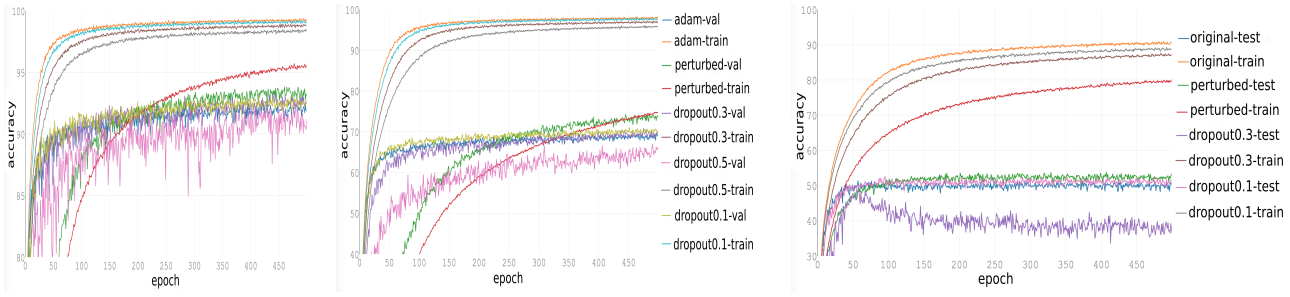
The right hand side of (3) has $\mathbb{E}_u[\hat{L}(w+u)]$. This suggests rather than minimizing the empirical loss $\hat{L}(w)$, we should optimize the perturbed empirical loss $\mathbb{E}_u[\hat{L}(w+u)]$ instead for a better model generalization power.

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 $\nabla^2\hat{L}$. The details of the algorithm is presented in Algorithm 1, where we treat η as a hyper-parameter.

Algorithm 1 Perturbed OPT

Require η, $\gamma = 0.1$, $\beta_1 = 0.999$, $\beta_2 = 0.1$, ϵ=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**
        $\rho[i] \leftarrow \frac{|h_t[i] - h_{t-1}[i]|}{\|w_t - w_{t-1}\|}$
        $\kappa[i] \leftarrow \frac{\gamma}{\log(1+epoch)}|w_{t-1}[i]| + \epsilon$
        $\sigma_i \leftarrow \min\left(\frac{1}{\log(1+epoch)\sqrt{\eta(h_t[i]+\rho[i]\cdot\kappa[i])}}, \kappa[i]\right) \cdot \mathbf{1}_{|g_t[i]|<\beta_2}$
      end if
      $u_t[i] \sim U(-\sigma_i, \sigma_i)$(sample a set of perturbations)
    end for
    $g_{t+1} \leftarrow \nabla_w \hat{L}_t(w_t + u_t)$ (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)
    $w_{t+1} \leftarrow \mathrm{OPT}(w_t)$ (update w using off-the-shell algorithms)
    $t \leftarrow t + 1$
  end for
end for

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



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

Figure 6: Training and validation accuracy of PertOPT and Dropout on CIFAR and tiny image net.

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.

9 Experiments

In this section we evaluate our PertOPT algorithm on three real-world image recognition datasets: CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and Tiny ImageNet[7]. We use the Wide-ResNet (Zagoruyko and Komodakis, 2018) 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 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 200. 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 compare the PertOPT algorithm against the original model as well as the one with dropout (Srivastava et al., 2014). Dropout can be viewed as multiplicative perturbation using Bernoulli random variables. It has already been widely used in almost every deep models. We present results using the exact same wide resnet architectures except the dropout layers are turned on or off. We report the accuracy with dropout rate of 0.0, 0.1, 0.3, and 0.5 on CIFAR-10 and CIFAR-100. For Tiny ImageNet we report the result with dropout rate being 0.0, 0.1, and 0.3. For our Pertubed OPT algorithm all the dropout layers are turned off.

Figure (6a), (6b), and (6c) show the accuracy versus epochs for training and validation in CIFAR-10, CIFAR-100, and Tiny ImageNet respectively. In Figure (6a) and (6b), Adam-train, and adam-val use the wide resnet model with 0 dropout rate. Perturbed-val and perturbed-train use the same wide resnet with 0 dropout rate, but add perturbation according to algorithm 1. It is pretty clear that with added dropout the validation/test accuracy gets boosted compared to the original method. For CIFAR-10, dropout rate 0.3 seems to work best compared to all the other dropout configurations. For CIFAR-100 and Tiny ImageNet, dropout 0.1 seems to work better. This may be due to the fact that CIFAR-10 has less training samples so more regularization is needed to prevent overfit.

For our PertOPT algorithm, we observe the effect with perturbation appears similar to regularization. With the perturbation, the accuracy on the training set tends to decrease, but the test on the validation set increases.

Although both PertOPT and dropout can be viewed as certain kind of regularization, in all experiments the PertOPT algorithm shows better performance on the validation/test data sets compared to the dropout methods. One possible explanation is maybe the PertOPT algorithm puts different levels of perturbation on different parameters according to the local smoothness structures, while only one dropout rate is set for all the parameters across the model.

10 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. To the best of our knowledge, this is the first work that integrate Hessian in the model generalization bound 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.

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

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

References

S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit regularization in deep matrix factorization. *CoRR*, abs/1905.13655, 2019.

P. L. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *Neural Information Processing Systems (NeurIPS)*, 2017.

Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 2018.

P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *International Conference on Learning Representations (ICLR)*, 2017.

L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. *International Conference on Machine Learning (ICML)*, 2017.

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Uncertainty in Artificial Intelligence (UAI)*, 2017.

P. Germain, F. Bach, A. Lacoste, and S. Lacoste-Julien. Pac-bayesian theory meets bayesian inference. *Conference on Neural Information Processing Systems (NIPS)*, 2016.

P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch SGD: training imagenet in 1 hour. *http://arxiv.org/abs/1706.02677*, 2017.

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

S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems 30*, 2017.

N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight VC-dimension bounds for piecewise linear neural networks. *Conference on Learning Theory (COLT)*, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *European Conference on Computer Vision (ECCV)*, 2014.

G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012.

E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *International Conference on Neural Information Processing Systems (NIPS)*, 2017.

G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

S. Jastrzębski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three factors influencing minima in sgd. *https://arxiv.org/abs/1711.04623*, 2017.

C. Jin, P. Netrapalli, and M. I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. *Conference On Learning Theory (COLT)*, 2018.

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2017.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2014.

A. Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems (NIPS)*, 2012.

J. Langford and R. Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems (NIPS)*, 2001.

J. Langford and J. Shawe-Taylor. Pac-bayes & margins. *International Conference on Neural Information Processing Systems (NIPS)*, 2002.

D. J. C. Mackay. Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Network Computation in Neural Systems*, 1995.

D. A. McAllester. Some pac-bayesian theorems. *Conference on Learning Theory (COLT)*, 1998.

D. A. McAllester. Pac-bayesian model averaging. *Conference on Learning Theory (COLT)*, 1999.

D. A. McAllester. Simplified pac-bayesian margin bounds. *Conference on Learning Theory (COLT)*, 2003.

B. McCann, N. S. Keskar, C. Xiong, and R. Socher. The natural language decathlon: Multitask learning as question answering. *https://arxiv.org/abs/1806.08730*, 2018.

A. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012.

Y. Nesterov and B. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 2006.

B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. *Conference on Neural Information Processing Systems (NIPS)*, 2017.

B. Neyshabur, S. Bhojanapalli, and N. Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations (ICLR)*, 2018.

R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *International Conference on Learning Representations (ICLR)*, 2018a.

R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *International Conference on Learning Representations (ICLR)*, 2018b.

Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. Pac-bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 2012a.

Y. Seldin, F. Laviolette, and J. Shawe-Taylor. Pac-bayesian analysis of supervised, unsupervised, and reinforcement learning. *International Conference on Machine Learning (Tutorials)*, 2012b.

S. L. Smith and Q. V. Le. A bayesian perspective on generalization and stochastic gradient descent. *International Conference on Learning Representations (ICLR)*, 2018.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

D. Soudry, E. Hoffer, and N. Srebro. The implicit bias of gradient descent on separable data. *International Conference on Learning Representations*, 2018.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Maching Learning Research*, 2014.

S. Zagoruyko and N. Komodakis. Wide residual networks. *British Machine Vision Conference (BMVC)*, 2018.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2016.

Z. Zhu, J. Wu, B. Yu, L. Wu, and J. Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from minima and regularization effects. *https://arxiv.org/abs/1803.00195*, 2018.