Implicit Bias/Regularization I

Instructor: Lei Wu [1]

Topics in Deep Learning Theory (Spring 2025)

Acknowledgements: This slide is prepared with the assistance of Zilin Wang.

[1] School of Mathematical Sciences; Center for Machine Learning Research

Table of Contents

“Modern” ML models are over-parameterized

Neural networks often work in the **over-parameterized** regime, i.e.,

$$\textbf{\# of parameters} \gg \textbf{\# of training samples.}$$

“Modern” ML models are over-parameterized

Neural networks often work in the **over-parameterized** regime, i.e.,

# of parameters $\gg$ # of training samples.

CIFAR-10	# train: 50,000
Inception	1,649,402
Alexnet	1,387,786
MLP 1x512	1,209,866
ImageNet	**# train: ~1,200,000**
Inception V4	42,681,353
Alexnet	61,100,840
Resnet-{18;152}	11,689,512; 60,192,808
VGG-{11;19}	132,863,336; 143,667,240

Figure 1: Different image classification models.

How to avoid overfitting?

- **Traditional-style ML:** Add *explicit regulatizations*, e.g., weight decay, batch/layer normalization, dropout, data argumentation.

How to avoid overfitting?

- **Traditional-style ML:** Add *explicit regulatizations*, e.g., weight decay, batch/layer normalization, dropout, data argumentation.
- **Modern ML:** A specific algorithm (with a specific initialization) only converges to certain solutions—A phenomenon referred to as implicit regularization/bias.

How to avoid overfitting?

- **Traditional-style ML:** Add *explicit regulatizations*, e.g., weight decay, batch/layer normalization, dropout, data argumentation.
- **Modern ML:** A specific algorithm (with a specific initialization) only converges to certain solutions—A phenomenon referred to as implicit regularization/bias.

Understanding **implicit regularization** is one of the most fundamental and mysterious problems in deep learning.

How to avoid overfitting?

- **Traditional-style ML:** Add *explicit regulatizations*, e.g., weight decay, batch/layer normalization, dropout, data argumentation.
- **Modern ML:** A specific algorithm (with a specific initialization) only converges to certain solutions—A phenomenon referred to as implicit regularization/bias.

Understanding **implicit regularization** is one of the most fundamental and mysterious problems in deep learning.

How to avoid overfitting?

- **Traditional-style ML:** Add *explicit regulatizations*, e.g., weight decay, batch/layer normalization, dropout, data argumentation.
- **Modern ML:** A specific algorithm (with a specific initialization) only converges to certain solutions—A phenomenon referred to as implicit regularization/bias.

Understanding **implicit regularization** is one of the most fundamental and mysterious problems in deep learning.

Key factors:

- Model
- Optimizer
- Initialization

Stochastic gradient descent

- Consider the empirical risk: $\hat{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$. Let $g_i(\theta) = \nabla\ell(f(x_i;\theta), y_i)$ and $g(\theta) = \frac{1}{n}\sum_i g_i(\theta)$.

Stochastic gradient descent

- Consider the empirical risk: $\hat{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$. Let $g_i(\theta) = \nabla \ell(f(x_i;\theta), y_i)$ and $g(\theta) = \frac{1}{n}\sum_i g_i(\theta)$.
- Gradient descent (GD): $\theta_{t+1} = \theta_t - \eta g(\theta_t)$.

Stochastic gradient descent

- Consider the empirical risk: $\hat{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i;\theta), y_i)$. Let $g_i(\theta) = \nabla \ell(f(x_i;\theta), y_i)$ and $g(\theta) = \frac{1}{n}\sum_i g_i(\theta)$.
- Gradient descent (GD): $\theta_{t+1} = \theta_t - \eta g(\theta_t)$.
- Stochastic gradient descent (SGD):

$$\theta_{t+1} = \theta_t - \eta \underbrace{\frac{1}{B}\sum_{j \in I_t} g_j(\theta_t)}_{\text{Stochastic grad.}}.$$

The SGD hyperparameters: learning rate (LR) η and batch size B.

Stochastic gradient descent

- Consider the empirical risk: $\hat{L}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(f(x_i;\theta), y_i)$. Let $g_i(\theta) = \nabla \ell(f(x_i;\theta), y_i)$ and $g(\theta) = \frac{1}{n}\sum_i g_i(\theta)$.
- Gradient descent (GD): $\theta_{t+1} = \theta_t - \eta g(\theta_t)$.
- Stochastic gradient descent (SGD):

$$\theta_{t+1} = \theta_t - \eta \underbrace{\frac{1}{B}\sum_{j\in I_t} g_j(\theta_t)}_{\text{Stochastic grad.}}.$$

  The SGD hyperparameters: learning rate (LR) η and batch size B.
- SGD = GD + noise:

$$\theta_{t+1} = \theta_t - \eta\left(g(\theta_t) + \frac{1}{\sqrt{B}}\xi_t\right),$$

  where

$$\mathbb{E}[\xi_t] = 0,\ \mathbb{E}[\xi_t\xi_t^T] = \frac{1}{n}\sum_{i=1}^n (g_i(\theta_t) - g(\theta_t))(g_i(\theta_t) - g(\theta_t))^T.$$

  The noise is state-dependent!

Implicit regularization of SGD

- SGD often converge to solutions that generalize well without needing any explicit regularization.
- Implicit regularization are even more important than explicit regularization (in deep learning).

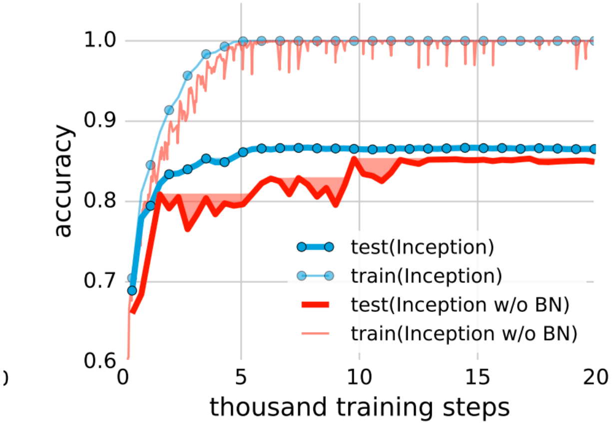


Figure 2: Classifying CIFAR-10 with Inception networks (Taken from [Chiyuan Zhang, et al, ICLR2017])

GD

GD can converge to good solutions.

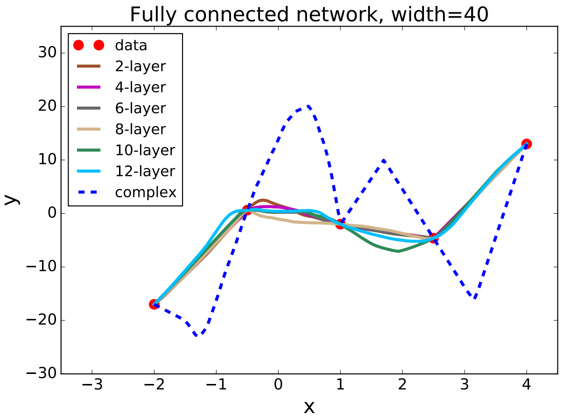


Figure 3: Taken from [Wu et al., 2018]. The dashed curve corresponds to bad solutions found by certain approach.

SGD performs better than GD

- **SGD often generalizes better than GD** although it is originally proposed to speed up training.

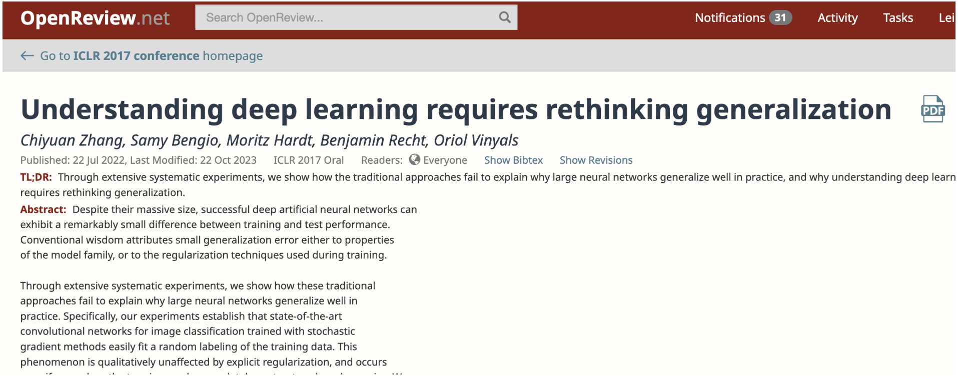
Experiment	Mini-batching	Epochs	Steps	Modifications	Val. Accuracy %
Baseline SGD	✓	300	117,000	-	95.70(±0.11)
Baseline FB	✗	300	300	-	75.42(±0.13)
FB train longer	✗	3000	3000	-	87.36(±1.23)
FB clipped	✗	3000	3000	clip	93.85(±0.10)
FB regularized	✗	3000	3000	clip+reg	95.36(±0.07)
FB strong reg.	✗	3000	3000	clip+reg+bs32	95.67(±0.08)
FB in practice	✗	3000	3000	clip+reg+bs32+shuffle	95.91(±0.14)

Table 2: Summary of validation accuracies in percent on the CIFAR-10 validation dataset for each of the experiments with data augmentations considered in Section 3. All validation accuracies are averaged over 5 runs.

Figure 4: Taken from (Geiping et al., ICLR 2022)

The ICLR 2017 Best Paper

- Understanding deep learning requires rethinking generalization by Chiyuan Zhang et al. Won the Best Paper Award of ICLR 2017.


OpenReview.net Search OpenReview... Notifications 31 Activity Tasks Lei

← Go to **ICLR 2017 conference** homepage

Understanding deep learning requires rethinking generalization

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, Oriol Vinyals

Published: 22 Jul 2022, Last Modified: 22 Oct 2023 ICLR 2017 Oral Readers: Everyone Show Bibtex Show Revisions

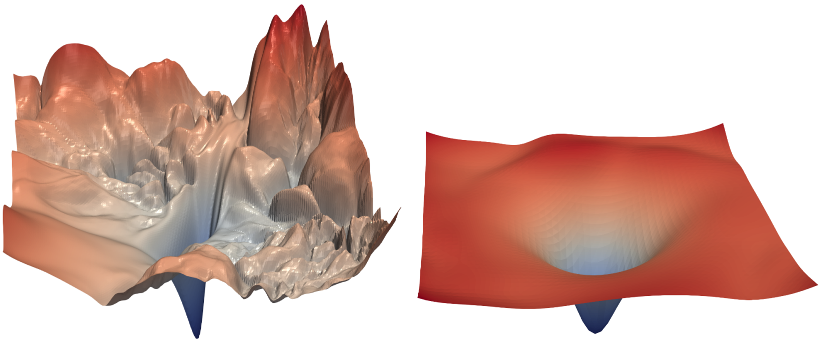
TL;DR: Through extensive systematic experiments, we show how the traditional approaches fail to explain why large neural networks generalize well in practice, and why understanding deep learn requires rethinking generalization.

Abstract: Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small difference between training and test performance. Conventional wisdom attributes small generalization error either to properties of the model family, or to the regularization techniques used during training.

Through extensive systematic experiments, we show how these traditional approaches fail to explain why large neural networks generalize well in practice. Specifically, our experiments establish that state-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data. This phenomenon is qualitatively unaffected by explicit regularization, and occurs

A Loss landscape Perspective

- The optimization of neural networks is highly non-convex. The loss landscape is usually full of global minima, bad local minima and saddle points.
  - Different minima have different local geometry.
  - The connectivity among different minima.
  - Many other topology and geometric structures.



(a) without skip connections (b) with skip connections

Figure 5: The loss surfaces of ResNet-56 without/with skip connections. [1]

Flat minima hypothesis (FMH)

The famous **flat minima hypothesis** (FMH):

1. SGD converges to flatter minima (Keskar et al., 2016).
2. Flatter minima generalize better (Hochreiter and Schmidhuber, 1995).

Flat minima hypothesis (FMH)

The famous **flat minima hypothesis** (FMH):

1. SGD converges to flatter minima (Keskar et al., 2016).
2. Flatter minima generalize better (Hochreiter and Schmidhuber, 1995).

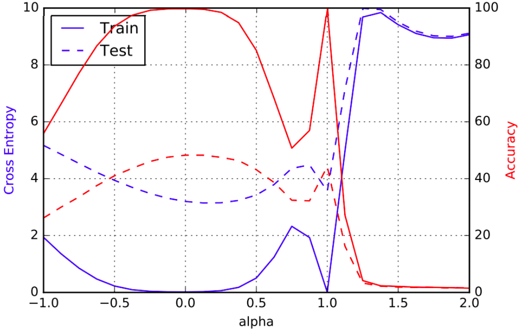


Figure 6: The landscape for $\theta(\alpha) := (1-\alpha)\theta_{SGD} + \alpha\theta_{GD}$. Taken from (Keskar et al., 2016).

Experiments in [2]

Table 1: Network Configurations

Name	Network Type	Architecture	Data set
F_1	Fully Connected	Section B.1	MNIST (LeCun et al., 1998a)
F_2	Fully Connected	Section B.2	TIMIT (Garofolo et al., 1993)
C_1	(Shallow) Convolutional	Section B.3	CIFAR-10 (Krizhevsky & Hinton, 2009)
C_2	(Deep) Convolutional	Section B.4	CIFAR-10
C_3	(Shallow) Convolutional	Section B.3	CIFAR-100 (Krizhevsky & Hinton, 2009)
C_4	(Deep) Convolutional	Section B.4	CIFAR-100

Table 2: Performance of small-batch (SB) and large-batch (LB) variants of ADAM on the 6 networks listed in Table 1

	Training Accuracy		Testing Accuracy	
Name	SB	LB	SB	LB
F_1	$99.66\% \pm 0.05\%$	$99.92\% \pm 0.01\%$	$98.03\% \pm 0.07\%$	$97.81\% \pm 0.07\%$
F_2	$99.99\% \pm 0.03\%$	$98.35\% \pm 2.08\%$	$64.02\% \pm 0.2\%$	$59.45\% \pm 1.05\%$
C_1	$99.89\% \pm 0.02\%$	$99.66\% \pm 0.2\%$	$80.04\% \pm 0.12\%$	$77.26\% \pm 0.42\%$
C_2	$99.99\% \pm 0.04\%$	$99.99\% \pm 0.01\%$	$89.24\% \pm 0.12\%$	$87.26\% \pm 0.07\%$
C_3	$99.56\% \pm 0.44\%$	$99.88\% \pm 0.30\%$	$49.58\% \pm 0.39\%$	$46.45\% \pm 0.43\%$
C_4	$99.10\% \pm 1.23\%$	$99.57\% \pm 1.84\%$	$63.08\% \pm 0.5\%$	$57.81\% \pm 0.17\%$

Experiments in [2] (Cont'd)

Table 4: Sharpness of Minima in Random Subspaces of Dimension 100

	$\epsilon = 10^{-3}$		$\epsilon = 5 \cdot 10^{-4}$	
	SB	LB	SB	LB
F_1	0.11 ± 0.00	9.22 ± 0.56	0.05 ± 0.00	9.17 ± 0.14
F_2	0.29 ± 0.02	23.63 ± 0.54	0.05 ± 0.00	6.28 ± 0.19
C_1	2.18 ± 0.23	137.25 ± 21.60	0.71 ± 0.15	29.50 ± 7.48
C_2	0.95 ± 0.34	25.09 ± 2.61	0.31 ± 0.08	5.82 ± 0.52
C_3	17.02 ± 2.20	236.03 ± 31.26	4.03 ± 1.45	86.96 ± 27.39
C_4	6.05 ± 1.13	72.99 ± 10.96	1.89 ± 0.33	19.85 ± 4.12

Remarks on Flat Minima Hypothesis

- In practice, FMH is very successful in guiding hyperparameter tuning and designing new optimizer for better generalization, e.g., the sharpness-aware minimization (Foret et al., 2021).

Remarks on Flat Minima Hypothesis

- In practice, FMH is very successful in guiding hyperparameter tuning and designing new optimizer for better generalization, e.g., the sharpness-aware minimization (Foret et al., 2021).
- However, FMH is only *empirically* validated. Theoretical understandings of the underlying mechanism is still limited.

Remarks on Flat Minima Hypothesis

- In practice, FMH is very successful in guiding hyperparameter tuning and designing new optimizer for better generalization, e.g., the sharpness-aware minimization (Foret et al., 2021).
- However, FMH is only *empirically* validated. Theoretical understandings of the underlying mechanism is still limited.

Remarks on Flat Minima Hypothesis

- In practice, FMH is very successful in guiding hyperparameter tuning and designing new optimizer for better generalization, e.g., the sharpness-aware minimization (Foret et al., 2021).
- However, FMH is only *empirically* validated. Theoretical understandings of the underlying mechanism is still limited.

Theoretical Questions:

- What is the appropriate metric of measuring "flatness"?

Remarks on Flat Minima Hypothesis

- In practice, FMH is very successful in guiding hyperparameter tuning and designing new optimizer for better generalization, e.g., the sharpness-aware minimization (Foret et al., 2021).
- However, FMH is only *empirically* validated. Theoretical understandings of the underlying mechanism is still limited.

Theoretical Questions:

- What is the appropriate metric of measuring "flatness"?
- Why does SGD prefer flat minima?

Remarks on Flat Minima Hypothesis

- In practice, FMH is very successful in guiding hyperparameter tuning and designing new optimizer for better generalization, e.g., the sharpness-aware minimization (Foret et al., 2021).
- However, FMH is only *empirically* validated. Theoretical understandings of the underlying mechanism is still limited.

Theoretical Questions:

- What is the appropriate metric of measuring "flatness"?
- Why does SGD prefer flat minima?
- Why does flat minima generalize well?

Sharpness Metrics

- Sharpness characterizes how the loss function (model prediction) changes under small perturbation in the parameter space. A sharp minimum corresponds to model, whose prediction is sensitive to the change of the parameters.

Sharpness Metrics

- Sharpness characterizes how the loss function (model prediction) changes under small perturbation in the parameter space. A sharp minimum corresponds to model, whose prediction is sensitive to the change of the parameters.
- The specific sharpness metric depends on what do we mean by **"small perturbation"**:

Sharpness Metrics

- Sharpness characterizes how the loss function (model prediction) changes under small perturbation in the parameter space. A sharp minimum corresponds to model, whose prediction is sensitive to the change of the parameters.
- The specific sharpness metric depends on what do we mean by **"small perturbation"**:
- At a minimum θ^*, we have

$$\lambda_{\max}(\nabla^2 L(\theta^*)) = \max_{\|\epsilon\|_2 \leq \rho} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

$$\mathrm{Tr}(\nabla^2 L(\theta^*)) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 I)} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

$$\|\nabla^2 L(\theta^*)\|_F^2 = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 \nabla^2 L(\theta^*))} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

Sharpness Metrics

- Sharpness characterizes how the loss function (model prediction) changes under small perturbation in the parameter space. A sharp minimum corresponds to model, whose prediction is sensitive to the change of the parameters.
- The specific sharpness metric depends on what do we mean by **"small perturbation"**:
- At a minimum θ^*, we have

$$\lambda_{\max}(\nabla^2 L(\theta^*)) = \max_{\|\epsilon\|_2 \le \rho} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

$$\mathrm{Tr}(\nabla^2 L(\theta^*)) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 I)} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

$$\|\nabla^2 L(\theta^*)\|_F^2 = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 \nabla^2 L(\theta^*))} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

- What happens if the perturbation is not very small? Does high-order gradient matter?

Sharpness Metrics

- Sharpness characterizes how the loss function (model prediction) changes under small perturbation in the parameter space. A sharp minimum corresponds to model, whose prediction is sensitive to the change of the parameters.
- The specific sharpness metric depends on what do we mean by **"small perturbation"**:
- At a minimum θ^*, we have

$$\lambda_{\max}(\nabla^2 L(\theta^*)) = \max_{\|\epsilon\|_2 \leq \rho} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$
$$\mathrm{Tr}(\nabla^2 L(\theta^*)) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 I)} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$
$$\|\nabla^2 L(\theta^*)\|_F^2 = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho^2 \nabla^2 L(\theta^*))} \frac{2(L(\theta^* + \epsilon) - L(\theta^*))}{\rho^2} + O(\rho)$$

- What happens if the perturbation is not very small? Does high-order gradient matter?
- Any connection with model's adversarial/random robustness?

Why Do Flat Minima Generalize Well?

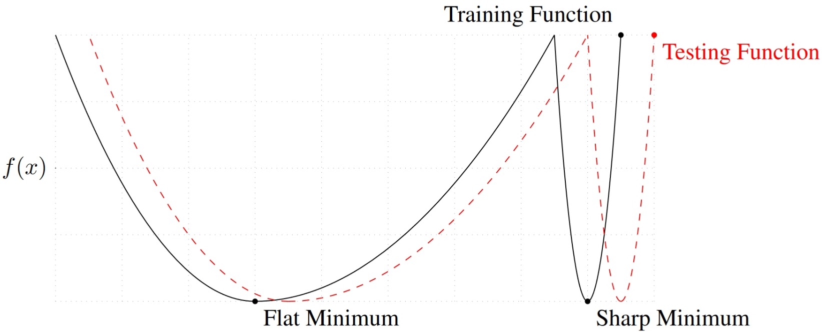


Figure 7: The most popular intuitive explanation of why flat minima generalize well provided in [2].

Why Do Flat Minima Generalize Well?

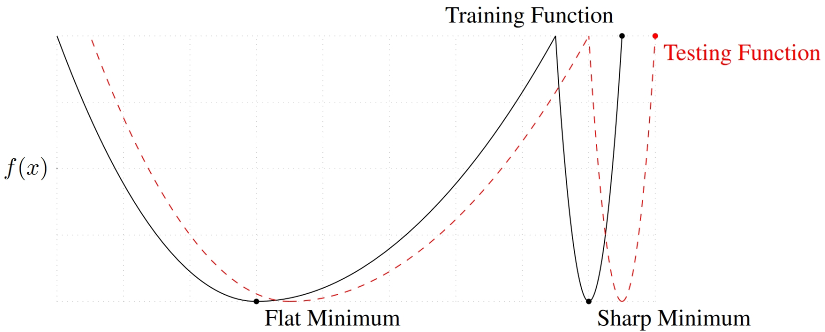


Figure 7: The most popular intuitive explanation of why flat minima generalize well provided in [2].

Remark:

- This illustration is "misleading" as it essentially suggests that sharp minima cannot generalize well! But this is wrong!
- This is due to in high dimensions, the testing landscape deviates the training landscape along flat directions .

Sharp Minima Can Generalize well

- ReLU networks are invariant under neural-wise rescaling:

$$a\sigma_{\mathrm{ReLU}}(\mathbf{w}^\top \mathbf{x}) = \frac{a}{\lambda}\sigma_{\mathrm{ReLU}}((\lambda \mathbf{w})^\top \mathbf{x}).$$

Sharp Minima Can Generalize well

- ReLU networks are invariant under neural-wise rescaling:

$$a\sigma_{\mathrm{ReLU}}(\mathbf{w}^\top \mathbf{x}) = \frac{a}{\lambda}\sigma_{\mathrm{ReLU}}((\lambda\mathbf{w})^\top \mathbf{x}).$$

- The rescaling operation $(a, \mathbf{w}) \to (a/\lambda, \lambda\mathbf{w})$ does not change the function implemented by the model, but can significantly change the sharpness.

Sharp Minima Can Generalize well

- ReLU networks are invariant under neural-wise rescaling:

$$a\sigma_{\mathrm{ReLU}}(\mathbf{w}^\top \mathbf{x}) = \frac{a}{\lambda}\sigma_{\mathrm{ReLU}}((\lambda\mathbf{w})^\top \mathbf{x}).$$

- The rescaling operation $(a, \mathbf{w}) \to (a/\lambda, \lambda\mathbf{w})$ does not change the function implemented by the model, but can significantly change the sharpness.
- Consider a toy landscape $L(a, w) = \frac{1}{2}(aw - 1)^2$. At global $\{(a, w) : aw = 1\}$, we have

$$\nabla^2 L(a, w) = \begin{pmatrix} w^2 & 1 \\ 1 & a^2 \end{pmatrix}.$$

Thus, rescaling can make a solution arbitrarily sharp.

Sharp Minima Can Generalize well

- ReLU networks are invariant under neural-wise rescaling:

$$a\sigma_{\mathrm{ReLU}}(\mathbf{w}^\top \mathbf{x}) = \frac{a}{\lambda}\sigma_{\mathrm{ReLU}}((\lambda \mathbf{w})^\top \mathbf{x}).$$

- The rescaling operation $(a, \mathbf{w}) \to (a/\lambda, \lambda \mathbf{w})$ does not change the function implemented by the model, but can significantly change the sharpness.
- Consider a toy landscape $L(a, w) = \frac{1}{2}(aw - 1)^2$. At global $\{(a, w) : aw = 1\}$, we have

$$\nabla^2 L(a, w) = \begin{pmatrix} w^2 & 1 \\ 1 & a^2 \end{pmatrix}.$$

  Thus, rescaling can make a solution arbitrarily sharp.
- This implies that we can **only expect flatness to be a sufficient condition for generalization**.

A PAC-Bayesian Perspective of FMH

- We will introduce the PAC-Bayesian explanation of FMH, which is the most popular theory in the community (Neyshabur et al., NIPS 2017).
- However, we will clarify that this explanation is very misleading.
- But **PAC-Bayes Theory itself is very useful**.

PAC-Bayes Theory: Setup

- Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. For brevity, denote by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ the joint space.

PAC-Bayes Theory: Setup

- Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. For brevity, denote by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ the joint space.
- Let $\mathcal{H}$ be the hypothesis/model space and $\mathcal{D} \in \mathcal{P}(\mathcal{Z})$ be the data distribution.

PAC-Bayes Theory: Setup

- Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. For brevity, denote by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ the joint space.
- Let $\mathcal{H}$ be the hypothesis/model space and $\mathcal{D} \in \mathcal{P}(\mathcal{Z})$ be the data distribution.
- Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}$ be the loss function. Then, $\ell(h, z)$ denotes the model h's prediction loss at z.

PAC-Bayes Theory: Setup

- Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. For brevity, denote by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ the joint space.
- Let $\mathcal{H}$ be the hypothesis/model space and $\mathcal{D} \in \mathcal{P}(\mathcal{Z})$ be the data distribution.
- Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}$ be the loss function. Then, $\ell(h, z)$ denotes the model h's prediction loss at z.
- Let $S = \{z_1, z_2, \ldots, z_n\}$ be the training set. Assume $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ and denote by $\hat{\mathcal{D}}_n = \frac{1}{n} \sum_{i=1}^{n} \delta(\cdot - z_i)$.

PAC-Bayes Theory: Setup

- Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. For brevity, denote by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ the joint space.
- Let $\mathcal{H}$ be the hypothesis/model space and $\mathcal{D} \in \mathcal{P}(\mathcal{Z})$ be the data distribution.
- Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}$ be the loss function. Then, $\ell(h, z)$ denotes the model h's prediction loss at z.
- Let $S = \{z_1, z_2, \ldots, z_n\}$ be the training set. Assume $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ and denote by $\hat{\mathcal{D}}_n = \frac{1}{n}\sum_{i=1}^n \delta(\cdot - z_i)$.
- Then, we denote by

$$L(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h, z)], \qquad \hat{L}(h) = \mathbb{E}_{z\sim\hat{\mathcal{D}}_n}[\ell(h, z)],$$

the population and empirical loss, respectively.

PAC-Bayes Theory: Setup

- Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, respectively. For brevity, denote by $\mathcal{Z} = \mathcal{X} \otimes \mathcal{Y}$ the joint space.
- Let $\mathcal{H}$ be the hypothesis/model space and $\mathcal{D} \in \mathcal{P}(\mathcal{Z})$ be the data distribution.
- Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}$ be the loss function. Then, $\ell(h, z)$ denotes the model h's prediction loss at z.
- Let $S = \{z_1, z_2, \ldots, z_n\}$ be the training set. Assume $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ and denote by $\hat{\mathcal{D}}_n = \frac{1}{n}\sum_{i=1}^n \delta(\cdot - z_i)$.
- Then, we denote by

$$L(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h, z)], \qquad \hat{L}(h) = \mathbb{E}_{z\sim\hat{\mathcal{D}}_n}[\ell(h, z)],$$

  the population and empirical loss, respectively.
- Consider a posterior distribution $Q \in \mathcal{P}(\mathcal{H})$ over the model space $\mathcal{H}$. Then, we can define generalization of Q by

$$L(Q) = \mathbb{E}_{h\sim Q}[L(h)], \qquad \hat{L}(Q) = \mathbb{E}_{h\sim Q}[\hat{L}(h)].$$

PAC-Bayesian Generalization Bound

Theorem 1 (McAllester (1998, 1999a))

Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto [0, 1]$ *be a loss function and* P *be a prior distribution over* $\mathcal{H}$. *Then, for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$ *over the sampling of* S, *we have for any* $Q \in \mathcal{P}(\mathcal{H})$, *it holds that*

$$L(Q) \leq \hat{L}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q||P) + \log(1/\delta)}{2n}}$$

PAC-Bayesian Generalization Bound

Theorem 1 (McAllester (1998, 1999a))

Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto [0,1]$ be a loss function and P be a prior distribution over $\mathcal{H}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of S, we have for any $Q \in \mathcal{P}(\mathcal{H})$, it holds that

$$L(Q) \leq \hat{L}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q||P) + \log(1/\delta)}{2n}}$$

Remark:

- The posterior distribution can Q can depend on the training set S but P cannot.

PAC-Bayesian Generalization Bound

Theorem 1 (McAllester (1998, 1999a))

Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto [0,1]$ be a loss function and P be a prior distribution over $\mathcal{H}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of S, we have for any $Q \in \mathcal{P}(\mathcal{H})$, it holds that

$$L(Q) \leq \hat{L}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q||P) + \log(1/\delta)}{2n}}$$

Remark:

- The posterior distribution can Q can depend on the training set S but P cannot.
- One often takes P, Q as certain Gaussian distributions since which the KL divergence between two Gaussian has an explicit form.

PAC-Bayesian Generalization Bound

Theorem 1 (McAllester (1998, 1999a))

Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto [0, 1]$ be a loss function and P be a prior distribution over $\mathcal{H}$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the sampling of S, we have for any $Q \in \mathcal{P}(\mathcal{H})$, it holds that

$$L(Q) \leq \hat{L}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q||P) + \log(1/\delta)}{2n}}$$

Remark:

- The posterior distribution can Q can depend on the training set S but P cannot.
- One often takes P, Q as certain Gaussian distributions since which the KL divergence between two Gaussian has an explicit form.
- PAC-Bayes theory has many application in machine learning theory. In this lecture, we will focus its application in explaining FMH.

PAC-Bayesian Generalization Bound

Theorem 1 (McAllester (1998, 1999a))

Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto [0,1]$ be a loss function and P be a prior distribution over $\mathcal{H}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of S, we have for any $Q \in \mathcal{P}(\mathcal{H})$, it holds that

$$L(Q) \leq \hat{L}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q||P) + \log(1/\delta)}{2n}}$$

Remark:

- The posterior distribution can Q can depend on the training set S but P cannot.
- One often takes P, Q as certain Gaussian distributions since which the KL divergence between two Gaussian has an explicit form.
- PAC-Bayes theory has many application in machine learning theory. In this lecture, we will focus its application in explaining FMH.
- We refer interested readers to [3, Chapter 31] and [4] for more materials about PAC-Bayes theory.

Donsker and Varadhan's Variational Principle

- Let $\mathcal{X}$ be general domain. Let $V : \mathcal{X} \mapsto \mathbb{R}$ be an (negative) energy function and $\pi \in \mathcal{P}(\mathcal{X})$ be an arbitrary underling distribution. Denote by π_V the corresponding Gibbs distribution given by
$$\frac{\mathrm{d}\pi_V}{\mathrm{d}\pi}(x) = \frac{e^{V(x)}}{\mathbb{E}_{x\sim\pi}[e^{V(x)}]}.$$

Donsker and Varadhan's Variational Principle

- Let $\mathcal{X}$ be general domain. Let $V : \mathcal{X} \mapsto \mathbb{R}$ be an (negative) energy function and $\pi \in \mathcal{P}(\mathcal{X})$ be an arbitrary underling distribution. Denote by π_V the corresponding Gibbs distribution given by

$$\frac{\mathrm{d}\pi_V}{\mathrm{d}\pi}(x) = \frac{e^{V(x)}}{\mathbb{E}_{x\sim\pi}[e^{V(x)}]}.$$

Theorem 2 (Donsker and Varadhan, 1976)

The Gibbs distribution satisfies

$$\pi_V = \operatorname*{argmax}_{p\in\mathcal{P}(\mathcal{X})} \left(\mathbb{E}_{x\sim p}[V(x)] - D_{\mathrm{KL}}(p||\pi)\right)$$

and moreover

$$\log \mathbb{E}_{x\sim\pi}[e^{V(x)}] = \sup_{p\in\mathcal{P}(\mathcal{X})} \left(\mathbb{E}_{x\sim p}[V(x)] - D_{\mathrm{KL}}(p||\pi)\right).$$

Donsker and Varadhan's Variational Principle

- Let $\mathcal{X}$ be general domain. Let $V : \mathcal{X} \mapsto \mathbb{R}$ be an (negative) energy function and $\pi \in \mathcal{P}(\mathcal{X})$ be an arbitrary underling distribution. Denote by π_V the corresponding Gibbs distribution given by
$$\frac{\mathrm{d}\pi_V}{\mathrm{d}\pi}(x) = \frac{e^{V(x)}}{\mathbb{E}_{x\sim\pi}[e^{V(x)}]}.$$

Theorem 2 (Donsker and Varadhan, 1976)

The Gibbs distribution satisfies
$$\pi_V = \operatorname*{argmax}_{p\in\mathcal{P}(\mathcal{X})} \left(\mathbb{E}_{x\sim p}[V(x)] - D_{\mathrm{KL}}(p||\pi)\right)$$
and moreover
$$\log \mathbb{E}_{x\sim\pi}[e^{V(x)}] = \sup_{p\in\mathcal{P}(\mathcal{X})} \left(\mathbb{E}_{x\sim p}[V(x)] - D_{\mathrm{KL}}(p||\pi)\right).$$

It is implied that for a given $p \in \mathcal{P}(\mathcal{X})$, it holds for any $\pi \in \mathcal{P}(\mathcal{X}), \lambda > 0$ that
$$\mathbb{E}_p[V] \le \log \mathbb{E}_\pi[e^V] + D_{\mathrm{KL}}(p||\pi) \quad \Longrightarrow \quad e^{\mathbb{E}_p[V]} \le \mathbb{E}_\pi[e^V]e^{D_{\mathrm{KL}}(p||\pi)}.$$

The blue one can be viewed as a generalized Jensen inequality.

Donsker and Varadhan's Variational Formula (Cont'd)

Proof:

$$\begin{aligned} D_{\mathrm{KL}}(p||\pi_V) &= \int \log\left(\frac{\mathrm{d}p}{\mathrm{d}\pi_V}\right)\mathrm{d}p \\ &= \int \log\left(\frac{\mathrm{d}p}{\mathrm{d}\pi}\frac{\mathrm{d}\pi}{\mathrm{d}\pi_V}\right)\mathrm{d}p \\ &= D_{\mathrm{KL}}(p||\pi) + \mathbb{E}_{x\sim p}\left[\log\left(\frac{\mathbb{E}[e^V]}{e^{V(x)}}\right)\right] \\ &= D_{\mathrm{KL}}(p||\pi) - \mathbb{E}_{x\sim p}[V(x)] + \log\mathbb{E}[e^V]. \end{aligned}$$

Then, the proof is completed by noticing $D_{\mathrm{KL}}(p||\pi_V) \geq 0$ and the equality is achieved when $p = \pi_V$.

Proof of the PAC-Bayesian Bound [2]

- Let $\widehat{\Delta}_n(h) = L(h) - \hat{L}(h)$ (generalization gap). Then, we need to bound $\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]$.

[2] A proof for bounding the expected generalization gap is more intuitive.

Proof of the PAC-Bayesian Bound [2]

- Let $\widehat{\Delta}_n(h) = L(h) - \hat{L}(h)$ (generalization gap). Then, we need to bound $\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]$.
- Recall Hoeffding's inequality: For any $\lambda > 0$,

$$\mathbb{E}_S[e^{\lambda \widehat{\Delta}_n(h)}] \leq e^{\frac{\lambda^2}{8n}}.$$

[2] A proof for bounding the expected generalization gap is more intuitive.

Proof of the PAC-Bayesian Bound [2]

- Let $\widehat{\Delta}_n(h) = L(h) - \hat{L}(h)$ (generalization gap). Then, we need to bound $\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]$.
- Recall Hoeffding's inequality: For any $\lambda > 0$,

$$\mathbb{E}_S[e^{\lambda \widehat{\Delta}_n(h)}] \le e^{\frac{\lambda^2}{8n}}.$$

- By the Chernoff-Cramer approach, we have

$$\mathbb{P}_S\left(\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \ge t\right) \ge \frac{\mathbb{E}_S[e^{\lambda\, \mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]}]}{e^{\lambda t}}$$

[2] A proof for bounding the expected generalization gap is more intuitive.

Proof of the PAC-Bayesian Bound [2]

- Let $\widehat{\Delta}_n(h) = L(h) - \hat{L}(h)$ (generalization gap). Then, we need to bound $\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]$.
- Recall Hoeffding's inequality: For any $\lambda > 0$,

$$\mathbb{E}_S[e^{\lambda \widehat{\Delta}_n(h)}] \leq e^{\frac{\lambda^2}{8n}}.$$

- By the Chernoff-Cramer approach, we have

$$\mathbb{P}_S\left(\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \geq t\right) \geq \frac{\mathbb{E}_S[e^{\lambda\,\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]}]}{e^{\lambda t}}$$

- By Donsker and Varadhan's variational principle,

$$e^{\lambda\,\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]} \leq \mathbb{E}_{h\sim P}[e^{\lambda \widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)}$$

[2] A proof for bounding the expected generalization gap is more intuitive.

Proof of the PAC-Bayesian Bound (Cont'd)

- Taking expectation wrt S gives

$$\begin{aligned}
\mathbb{E}_S\big[e^{\lambda\,\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]}\big] &\le \mathbb{E}_S\,\mathbb{E}_{h\sim P}\big[e^{\lambda\widehat{\Delta}_n(h)}\big]e^{D_{\mathrm{KL}}(Q||P)} \\
&\le \mathbb{E}_{h\sim P}\,\mathbb{E}_S\big[e^{\lambda\widehat{\Delta}_n(h)}\big]e^{D_{\mathrm{KL}}(Q||P)} \\
&\le e^{\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}.
\end{aligned}$$

Proof of the PAC-Bayesian Bound (Cont'd)

- Taking expectation wrt S gives

$$\begin{aligned}
\mathbb{E}_S[e^{\lambda \mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]}] &\leq \mathbb{E}_S \mathbb{E}_{h\sim P}[e^{\lambda \widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)} \\
&\leq \mathbb{E}_{h\sim P} \mathbb{E}_S[e^{\lambda \widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)} \\
&\leq e^{\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}.
\end{aligned}$$

- Therefore, we have

$$\mathbb{P}_S\left(\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \geq t\right) \leq e^{-\lambda t+\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}$$

Proof of the PAC-Bayesian Bound (Cont'd)

- Taking expectation wrt S gives

$$\begin{aligned}
\mathbb{E}_S[e^{\lambda\,\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]}] &\leq \mathbb{E}_S\,\mathbb{E}_{h\sim P}[e^{\lambda\widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)} \\
&\leq \mathbb{E}_{h\sim P}\,\mathbb{E}_S[e^{\lambda\widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)} \\
&\leq e^{\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}.
\end{aligned}$$

- Therefore, we have

$$\mathbb{P}_S\left(\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \geq t\right) \leq e^{-\lambda t+\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}$$

- This yield with probability at least $1-\delta$, we have

$$\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \leq \frac{\lambda}{8n} + \frac{D_{\mathrm{KL}}(Q||P)+\log(1/\delta)}{\lambda}.$$

Proof of the PAC-Bayesian Bound (Cont'd)

- Taking expectation wrt S gives

$$\begin{aligned}
\mathbb{E}_S[e^{\lambda\,\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)]}] &\leq \mathbb{E}_S\,\mathbb{E}_{h\sim P}[e^{\lambda\widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)} \\
&\leq \mathbb{E}_{h\sim P}\,\mathbb{E}_S[e^{\lambda\widehat{\Delta}_n(h)}]e^{D_{\mathrm{KL}}(Q||P)} \\
&\leq e^{\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}.
\end{aligned}$$

- Therefore, we have

$$\mathbb{P}_S\left(\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \geq t\right) \leq e^{-\lambda t+\frac{\lambda^2}{8n}+D_{\mathrm{KL}}(Q||P)}$$

- This yield with probability at least $1-\delta$, we have

$$\mathbb{E}_{h\sim Q}[\widehat{\Delta}_n(h)] \leq \frac{\lambda}{8n} + \frac{D_{\mathrm{KL}}(Q||P)+\log(1/\delta)}{\lambda}.$$

- Optimizing λ completes the proof.

PAC-Bayesian Generalization Bound for Flat Minima

Theorem 1 (PAC-Bayesian bound for sharpness-generalization, [5])

Suppose $\ell : \Theta \times \mathcal{Z} \mapsto [0,1]$. For any $\rho > 0$, if we assume $L(\theta) \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[L(\theta+\epsilon)]$, then w.p. at least $1-\delta$ over the choice of $\mathcal{S}$, we have

$$L(\theta) - \hat{L}(\theta) \leq \underbrace{\max_{\|\epsilon\|_2 \leq \sigma} \hat{L}(\theta+\epsilon) - \hat{L}(\theta)}_{\text{Sharpness}} + \sqrt{\frac{k \log\left(1 + \frac{\|\theta\|_2^2}{\rho^2}\left(1+\sqrt{\frac{\log n}{k}}\right)\right) + 4\log\frac{n}{\delta} + \tilde{O}(1)}{n}}$$

where k is the number of parameter space.

PAC-Bayesian Generalization Bound for Flat Minima

Theorem 1 (PAC-Bayesian bound for sharpness-generalization, [5])

Suppose $\ell : \Theta \times \mathcal{Z} \mapsto [0, 1]$. For any $\rho > 0$, if we assume $L(\theta) \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)}[L(\theta + \epsilon)]$, then w.p. at least $1 - \delta$ over the choice of $\mathcal{S}$, we have

$$L(\theta) - \hat{L}(\theta) \leq \underbrace{\max_{\|\epsilon\|_2 \leq \sigma} \hat{L}(\theta + \epsilon) - \hat{L}(\theta)}_{\text{Sharpness}} + \sqrt{\frac{k \log\left(1 + \frac{\|\theta\|_2^2}{\rho^2}\left(1 + \sqrt{\frac{\log n}{k}}\right)\right) + 4 \log \frac{n}{\delta} + \tilde{O}(1)}{n}}$$

where k is the number of parameter space.

- **Let us criticize this "theorem"!**

Proof

- Let $Q = \mathcal{N}(\theta, \sigma^2 I_k)$. Then, by PAC-Bayesian bound, we have

$$L(\theta) \leq \mathbb{E}_{\theta \sim Q}[L(\theta)] \leq \mathbb{E}_{\theta \sim Q}[\hat{L}(\theta)] + \sqrt{\frac{D_{\mathrm{KL}}(Q \| P) + \log \frac{1}{\delta}}{2n}}$$

[3] This is not correct since σ_P^2 should not depend on the training set S. Fortunately, this issue can be fixed by a standard union bound argument. We leave this to homework.

Proof

- Let $Q = \mathcal{N}(\theta, \sigma^2 I_k)$. Then, by PAC-Bayesian bound, we have

$$L(\theta) \leq \mathbb{E}_{\theta \sim Q}[L(\theta)] \leq \mathbb{E}_{\theta \sim Q}[\hat{L}(\theta)] + \sqrt{\frac{D_{\mathrm{KL}}(Q \| P) + \log \frac{1}{\delta}}{2n}}$$

- Recall that

$$D_{\mathrm{KL}}(\mathcal{N}(\mu_1, \Sigma_1) || \mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\left(\log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - k + (\mu_1 - \mu_2)\Sigma_2^{-1}(\mu_1 - \mu_2) + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1)\right)$$

[3] This is not correct since σ_P^2 should not depend on the training set S. Fortunately, this issue can be fixed by a standard union bound argument. We leave this to homework.

Proof

- Let $Q = \mathcal{N}(\theta, \sigma^2 I_k)$. Then, by PAC-Bayesian bound, we have

$$L(\theta) \leq \mathbb{E}_{\theta \sim Q}[L(\theta)] \leq \mathbb{E}_{\theta \sim Q}[\hat{L}(\theta)] + \sqrt{\frac{D_{\mathrm{KL}}(Q \| P) + \log \frac{1}{\delta}}{2n}}$$

- Recall that

$$D_{\mathrm{KL}}(\mathcal{N}(\mu_1, \Sigma_1) \| \mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\left(\log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - k + (\mu_1 - \mu_2)\Sigma_2^{-1}(\mu_1 - \mu_2) + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1)\right)$$

- Taking $P = \mathcal{N}(\mu_P, \sigma_P^2 I)$ gives,

$$D_{\mathrm{KL}}(Q \| P) = \frac{1}{2}\left[\frac{k\sigma^2 + \|\mu_P - \theta\|_2^2}{\sigma_P^2} - k + k \log\left(\frac{\sigma_P^2}{\sigma^2}\right)\right]$$

[3] This is not correct since σ_P^2 should not depend on the training set S. Fortunately, this issue can be fixed by a standard union bound argument. We leave this to homework.

Proof

- Let $Q = \mathcal{N}(\theta, \sigma^2 I_k)$. Then, by PAC-Bayesian bound, we have

$$L(\theta) \le \mathbb{E}_{\theta\sim Q}[L(\theta)] \le \mathbb{E}_{\theta\sim Q}[\hat{L}(\theta)] + \sqrt{\frac{D_{\mathrm{KL}}(Q\|P) + \log\frac{1}{\delta}}{2n}}$$

- Recall that

$$D_{\mathrm{KL}}(\mathcal{N}(\mu_1, \Sigma_1)\|\mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\left(\log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - k + (\mu_1 - \mu_2)\Sigma_2^{-1}(\mu_1 - \mu_2) + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1)\right)$$

- Taking $P = \mathcal{N}(\mu_P, \sigma_P^2 I)$ gives,

$$D_{\mathrm{KL}}(Q\|P) = \frac{1}{2}\left[\frac{k\sigma^2 + \|\mu_P - \theta\|_2^2}{\sigma_P^2} - k + k\log\left(\frac{\sigma_P^2}{\sigma^2}\right)\right]$$

- Taking $\mu_P = 0, \sigma_P^2 = \sigma^2 + k^{-1}\|\theta\|_2^2$ **nearly** [3] completes the proof.

---

[3] This is not correct since σ_P^2 should not depend on the training set S. Fortunately, this issue can be fixed by a standard union bound argument. We leave this to homework.

Remarks

The lessons what we learn from the aforementioned analysis include

- Flatness can be only a sufficient condition for generalization.
- Whether flat minima generalize depends on
  - Model architecture
  - Flatness metric
  - Data distribution.

An Important Observation

Consider the regression problem

$$\widehat{L}(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(x_i;\theta) - y_i)^2 =: \frac{1}{2n}\sum_{i=1}^{n} e_i^2.$$

- the Hessian

$$H(\theta) := \nabla^2 \widehat{L}(\theta) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\nabla f(x_i;\theta)\nabla f(x_i;\theta)^{\mathrm{T}}}_{G(\theta)} + \frac{1}{n}\sum_{i=1}^{n} e_i \nabla^2 f(x_i;\theta),$$

  where we refer to $G(\theta)$ as the empirical Fisher matrix.
- When $\widehat{L}(\theta)$ is small, we have $H(\theta) \approx G(\theta)$ and particularly, at **an interpolation minimum** θ^* where $\widehat{L}(\theta^*) = 0$, we have

$$H(\theta^*) = G(\theta^*)$$

- We shall measure the "sharpness" by using $G(\theta)$ instead of $H(\theta)$, e.g.,

$$\mathrm{Tr}[G(\theta)] = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f(x_i;\theta)\|^2.$$

Two-layer Networks (Without Bias)

- Consider two-layer ReLU networks (without bias) given by

$$f(x,\theta) = \sum_{j=1}^{m} a_j \sigma({w_j}^{\mathrm{T}} x)$$

where $a_j \in \mathbb{R}, w_j \in \mathbb{R}^d$ and $\sigma(t) = \max(t,0)$.

- We assume $x \sim \rho = \mathrm{Unif}(\sqrt{d}\mathbb{S}^{d-1})$.
- A simple calculation:

$$\begin{aligned}\mathrm{Tr}[G(\theta)] = \mathbb{E}_x[\|\nabla f(x;\theta)\|^2] &= \sum_{j=1}^{m} \left(\mathbb{E}[\sigma(w_j^\top x)^2] + a_j^2\, \mathbb{E}[|\sigma'(w_j^\top x)|^2 \|x\|^2]\right) \\ &= \sum_{j=1}^{m} (\gamma_1 \|w_j\|^2 + \gamma_2 d a_j^2),\end{aligned}$$

where γ_1, γ_2 are two absolute constants given by

$$\gamma_1 = \mathbb{E}_x[\sigma(x_1)^2], \qquad \gamma_2 = \mathbb{E}_x[\sigma'(x_1)^2].$$

Two-layer Networks (Cont'd)

Define a weight ℓ_2 norm as follows

$$\|\theta\|_{2,q} := \sqrt{\sum_{j=1}^{m}(\|w_j\|^2 + q a_j^2)}.$$

Theorem 2 (Thm 4.1 in [6])

Let $N(d,\delta) = \inf\{n \in \mathbb{N} : d\log(n/\delta)/n \leq 1\}$.

- If $n \gtrsim N(d,\delta)$, then it holds w.p. $1-\delta$ that

$$\mathrm{Tr}(G(\theta)) \sim \|\theta\|_{2,d} \qquad \|G(\theta)\|_F \sim \|\theta\|_{2,\sqrt{d}}.$$

- If $n \gtrsim dN(d,\delta)$, then it holds w.p. at least $1-\delta$ that

$$\|G(\theta)\|_2 \sim \|\theta\|_{2,1}.$$

Remark: It is worth noting that "sharpness" depends on the training data but the parameter norms do not!

How do we kill the data dependence?

Derivation on the backboard!

Flatness Implies Generalization

For ReLU networks, the generalization gap can be controlled by the path norm

$$\|\theta\|_{\mathcal{P}} := \sum_{j=1}^{m} |a_j| \|w_j\|$$

, which can be further upper bounded by the weighted ℓ_2 norm:

$$\|\theta\|_{2,q} = \sum_{j=1}^{m} (\|w_j\|^2 + q a_j^2) \geq 2\sqrt{q} \sum_{j=1}^{m} |a_j| \|w_j\| = 2\sqrt{q} \|\theta\|_{\mathcal{P}}.$$

Theorem 3 (Thm 4.3 in [6])

Suppose $\sup_x |f^*(x)| \leq 1$. For any $\delta \in (0,1)$, if $n \gtrsim N(d,\delta)$, then it holds w.p. at least $1-\delta$ for any interpolation minimum $\widehat{\theta}$ that

$$\mathbb{E}_x \|f(x;\hat{\theta}) - f^*(x)\|^2 \lesssim \frac{\operatorname{Tr}^2(G(\hat{\theta}))}{n} \operatorname{poly}(n, 1/\delta).$$

Remark:

- Similar results also hold for other metrics of sharpness. We show next that **a slight change of input distribution can cause that flat minima generalize poorly**.

Two-layer Networks With Bias

[7] shows that a slight modification of the input distribution causes that flat minima don't necessarily generalize.

- Data distribution: $x \sim \mathrm{Unif}(\{\pm 1\}^d)$ and $y = x^{(1)} x^{(2)}$.
- Model: Two-layer ReLU network with bias: $f(x, \theta) = \sum_{j=1}^{m} a_j \sigma(w_j^{\mathrm{T}} x + b_j)$.

Theorem 4 (Flat minima do not generalize, Theorem 4.1 in [7])

Under the setting above, if $m \geq n$, **there is a flattest global minimum that cannot generalize at all**. ("Flattest" is in the sense of Hessian trace, in terms of all global minima, i.e. $f(x_i, \theta) = y_i, \forall i$.)

Two-layer Networks With Bias (Cont'd)

Proof: Step 1: Construct a so-called memorizing solution.

Definition 1 (Memorizing solution)

A 2-layer network is a memorizing solution if (1) it interpolates the training dataset, i.e. global minimum, (2) any x_i in the training activates only one neuron in the hidden layer, and different x_i's activate different neurons.

- WLOG, assume $m = n$. For $j = 1, 2, \ldots, m$, let

$$w_j = x_j/\|x_j\|, \quad b_j = -0.5\sqrt{d} \quad a_j = y_j/(0.5\sqrt{d}).$$

- w.p. $1-\delta$, $\sup_{i,j\in[n]} |\hat{x}_i^T \hat{x}_j| \le \frac{\log(n/\delta)}{n}$.
- By the above choice, w.p. $1-\delta$, each sample can only activate one neuron. Consequently,

$$f(x_i;\tilde{\theta}) = \sum_{j=1}^{n} a_j \sigma(\hat{x}_j^\top x_i - 0.5\sqrt{d}) = a_i \sigma(\hat{x}_i^\top x_i - 0.5\sqrt{d}) = y_i.$$

- In this way we obtain a memorizing solution that predicts 0 anywhere outside the training set, thus no generalization at all. Or, to be specific, the generalization error is $1 - n/2^d$.

Two-layer Networks With Bias (Cont'd)

Step 2: Show that the memorizing solution is the flattest among all global minima.

- We do this by lower bounding the sharpness. Note that we still have

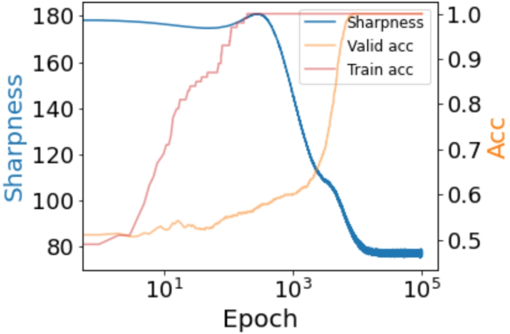
$$\mathrm{Tr}[G(\theta)] = \frac{1}{n}\sum_{i=1}^{n} \|\nabla_\theta f(x_i, \theta)\|^2$$

- For any x_i, we have $f(x_i, \theta) = \sum_{j=1}^{m} a_j \sigma(w_j^{\mathrm{T}} x_i + b_j) = y_i$. For simplicity of writing we introduce the new notations $w_j' = \mathrm{concat}(w_j, b_j) \in \mathbb{R}^{d+1}$ and $x_i' = \mathrm{concat}(x_j, 1) \in \mathbb{R}^{d+1}$. Then by Cauchy-Schwarz inequality,

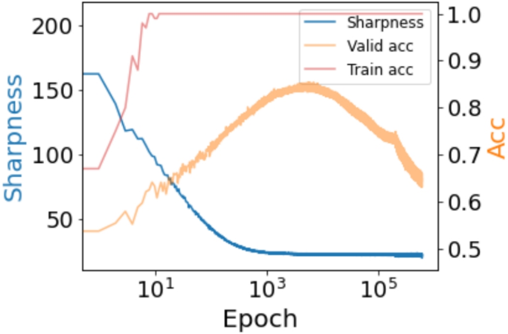
$$\begin{aligned}
\|\nabla_\theta f(x_i, \theta)\|^2 &= \sum_{j=1}^{m} \left(\sigma^2({w_j'}^{\mathrm{T}} x_i') + \|a_j \mathbb{1}({w_j'}^{\mathrm{T}} x_i' \geq 0) x_i'\|^2 \right) \\
&\geq \sum_{j=1}^{m} 2\sigma({w_j'}^{\mathrm{T}} x_i') |a_j| \mathbb{1}({w_j'}^{\mathrm{T}} x_i' \geq 0) \|x_i'\| \\
&\geq \left| \sum_{j=1}^{m} 2 a_j \sigma({w_j'}^{\mathrm{T}} x_i') \right| \|x_i'\| = 2\|x_i'\| |y_i|
\end{aligned}$$

We can choose an appropriate memorizing solution such that all equalities hold simultaneously. Therefore it is the flattest. □

Two-layer networks with bias (Cont'd)



(a) Baseline



(b) 1-SAM

Figure 8: FMH cannot explain the implicit regularization of SGD. The sharpness-aware minimization (SAM) find flatter solutions but they generalize worse.

Sharpness-Aware Minimization (SAM)

- Since we believe reducing sharpness can be helpful in generalization, **can we use this observation to design an algorithm with better generalization**?

Sharpness-Aware Minimization (SAM)

- Since we believe reducing sharpness can be helpful in generalization, **can we use this observation to design an algorithm with better generalization**?
- [5] proposes a sharpness-aware minimization (SAM), which aims to minimize

$$L^{\text{SAM}}(\theta) := \underbrace{L(\theta)}_{\text{fitting loss}} + \underbrace{\max_{\|\epsilon\|_2 \leq \rho} L(\theta + \epsilon) - L(\theta)}_{\text{sharpness}}$$

The "Unreasonable" Simplification

- However, the new loss is hard to calculate. Fortunately, we have the following approximation of the maximizer $\epsilon^*(\theta)$:

$$\epsilon^*(\theta) = \underset{\|\epsilon\|_2 \le \rho}{\operatorname{argmax}} L(\theta + \epsilon) \approx \underset{\|\epsilon\|_2 \le \rho}{\operatorname{argmax}} L(\theta) + \epsilon^{\mathrm{T}} \nabla_\theta L(\theta) = \rho \frac{\nabla_\theta L(\theta)}{\|\nabla_\theta L(\theta)\|_2} =: \epsilon(\theta)$$

- And the derivative

$$\nabla_\theta L^{SAM}(\theta) \approx \nabla_\theta L(\theta + \epsilon(\theta)) = \frac{\mathrm{d}(\theta + \epsilon(\theta))}{\mathrm{d}\theta} \nabla_\theta L(\theta)|_{\theta+\epsilon(\theta)} \approx \nabla_\theta L(\theta)|_{\theta+\epsilon(\theta)}$$

  in the last approximation we neglect the derivative of $\epsilon(\theta)$.

- In a summary, one SAM step goes like

$$\theta_{t+1} = \theta_t - \eta \nabla L \left(\theta_t + \rho \frac{\nabla L(\theta_t)}{\|\nabla L(\theta_t)\|} \right) \tag{1}$$

The Performance of SAM on Vision Tasks

Model	#params	Throughput (img/sec/core)	ImageNet	ReaL
			ResNet	
ResNet-50-SAM	25M	2161	76.7 (+0.7)	83.1 (+0.7)
ResNet-101-SAM	44M	1334	78.6 (+0.8)	84.8 (+0.9)
ResNet-152-SAM	60M	935	79.3 (+0.8)	84.9 (+0.7)
ResNet-50x2-SAM	98M	891	79.6 (+1.5)	85.3 (+1.6)
ResNet-101x2-SAM	173M	519	80.9 (+2.4)	86.4 (+2.4)
ResNet-152x2-SAM	236M	356	81.1 (+1.8)	86.4 (+1.9)
			Vision Transformer	
ViT-S/32-SAM	23M	6888	70.5 (+2.1)	77.5 (+2.3)
ViT-S/16-SAM	22M	2043	78.1 (+3.7)	84.1 (+3.7)
ViT-S/14-SAM	22M	1234	78.8 (+4.0)	84.8 (+4.5)
ViT-S/8-SAM	22M	333	81.3 (+5.3)	86.7 (+5.5)
ViT-B/32-SAM	88M	2805	73.6 (+4.1)	80.3 (+5.1)
ViT-B/16-SAM	87M	863	79.9 (+5.3)	85.2 (+5.4)

Figure 9: Table 2 in Chen, et al., (2022).

The Performance of SAM on NLP tasks

Model	SGlue	BoolQ	CB	CoPA	MultiRC	ReCoRD	RTE	WiC	WSC
Small	67.7	72.6	89.4 / **89.3**	**67.0**	**68.5** / 21.4	61.7 / 60.8	69.3	65.4	72.1
Small + SAM (0.05)	**68.4**	**73.5**	**92.1 / 89.3**	61.0	**68.5 / 22.8**	**62.1 / 61.0**	**69.7**	**65.7**	**79.8**
Base	75.3	80.0	91.7 / **94.6**	71.0	75.4 / 35.4	76.2 / 75.4	80.9	69.3	76.9
Base + SAM (0.15)	**78.5**	**82.2**	**93.7 / 94.6**	**78.0**	**77.5 / 39.1**	**78.2 / 77.2**	**85.9**	**70.4**	**81.7**
Large	84.3	86.6	**99.4 / 98.2**	**89.0**	83.7 / 51.0	86.5 / 85.6	**89.2**	72.9	84.6
Large + SAM (0.15)	**84.6**	**88.0**	95.0 / 96.4	86.0	**84.0 / 53.7**	**87.3 / 86.4**	**89.2**	**75.2**	**86.5**
XL	87.2	88.6	93.7 / 96.4	**95.0**	86.9 / 61.1	89.5 / 88.4	91.3	74.9	89.4
XL + SAM (0.15)	**89.1**	**89.4**	**100.0 / 100.0**	**95.0**	**87.9 / 63.7**	**90.9 / 90.0**	**92.1**	**75.5**	**94.2**

Table 1: Experimental results (dev scores) on the (full) SuperGLUE benchmark. Public checkpoints of various sizes are fine-tuned with and without SAM for 250k steps. We see that SAM improves performance across *all* model sizes.

Figure 10: Table 1 in Bahri, et al., (2022).

Remark: Small (77M), Base (250M), Large (880M), and XL (3B).

How SAM works?

- Let us take a look at the SAM update:

$$\theta_{t+1} = \theta_t - \eta \nabla L \left(\theta_t + \rho \frac{\nabla L(\theta_t)}{\|\nabla L(\theta_t)\|} \right)$$

- No explicit regularization at all. It should be certain implicit bias that improves the performance.
- How to formulate the implicit bias of SAM?

A Stability Perspective

The Escape Phenomenon

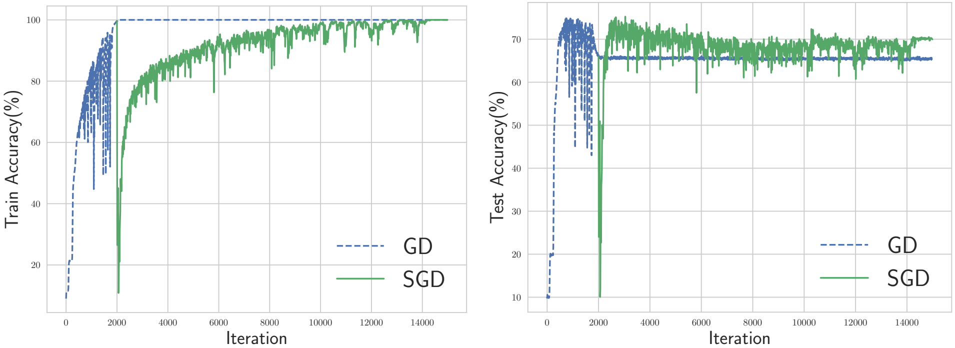


Figure 11: Fast escape phenomenon in fitting corrupted FashionMNIST.

Observation:

- This escape phenomenon indicates that GD solutions are **dynamically unstable** for SGD.

The Escape Phenomenon

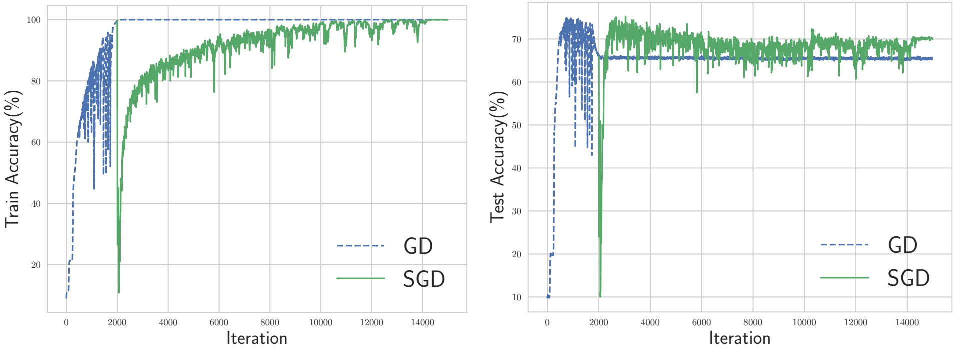


Figure 11: Fast escape phenomenon in fitting corrupted FashionMNIST.

Observation:

- This escape phenomenon indicates that GD solutions are **dynamically unstable** for SGD.
- The escape is **unreasonably fast**, providing a indicator of how much SGD dislikes sharp minima.

Stability of Gradient Flow

The gradient flow (GF) is GD with a infinite-small learning rate.

$$\dot{\theta}_t = -\nabla \hat{L}(\theta_t).$$

- All critical points ($\nabla \hat{L}(\theta) = 0$) are the fixed points of GF.
- But GF only prefers **minima** which are the stable ones. Saddle points are unstable; minima are stable.

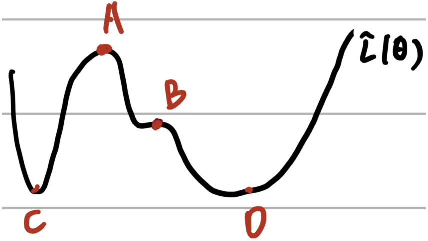


Figure 12: GF only selects C and D. A and B are unstable for GF.

Stability of Gradient Descent

Gradient descent (GD) updates as $\theta_{t+1} = \theta_t - \eta \nabla \hat{L}(\theta_t)$.

- GD with a large LR only converges to the minimum D.
- GD escape from the minimum C exponentially fast.

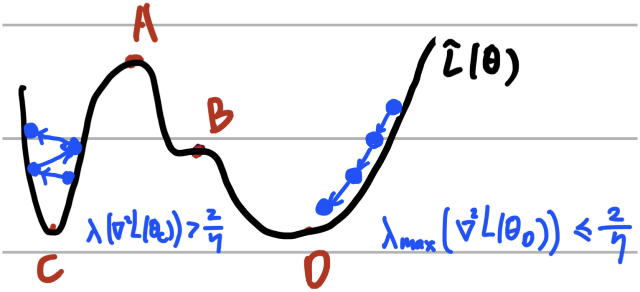


Figure 13: GD with a large LR only selects D. The minimum C is stable for GF but not for GD with a relatively large LR.

The Linear Stability Analysis

- **Linearize the GD dynamics:** Then, linearzing GD around θ^* gives

$$\begin{aligned}\theta_{t+1} - \theta^* &= \theta_t - \theta^* - \eta(\nabla L(\theta_t) - \nabla L(\theta^*)) \\ &\approx (I - \eta H(\theta^*))(\theta_t - \theta^*) \\ &= \color{blue}{(I - \eta H(\theta^*))^t}(\theta_0 - \theta^*).\end{aligned}$$

The Linear Stability Analysis

- **Linearize the GD dynamics:** Then, linearzing GD around θ^* gives

$$\begin{aligned}\theta_{t+1} - \theta^* &= \theta_t - \theta^* - \eta(\nabla L(\theta_t) - \nabla L(\theta^*)) \\ &\approx (I - \eta H(\theta^*))(\theta_t - \theta^*) \\ &= (I - \eta H(\theta^*))^t(\theta_0 - \theta^*).\end{aligned}$$

- **Stability condition:** Stability $\Rightarrow \|I - \eta H(\theta^*)\|_2 \leq 1 \Rightarrow$

$$\underbrace{\lambda_1(H(\theta^*))}_{\text{Sharpness}} \leq \frac{2}{\eta}.$$

Otherwise, GD escapes from that minimum exponentially fast: $(1 - \eta\lambda_1(H(\theta^*)))^t$.

The Linear Stability Analysis

- **Linearize the GD dynamics:** Then, linearzing GD around θ^* gives

$$
\begin{aligned}
\theta_{t+1} - \theta^* &= \theta_t - \theta^* - \eta(\nabla L(\theta_t) - \nabla L(\theta^*)) \\
&\approx (I - \eta H(\theta^*))(\theta_t - \theta^*) \\
&= (I - \eta H(\theta^*))^t(\theta_0 - \theta^*).
\end{aligned}
$$

- **Stability condition:** Stability $\Rightarrow \|I - \eta H(\theta^*)\|_2 \leq 1 \Rightarrow$

$$
\underbrace{\lambda_1(H(\theta^*))}_{\text{Sharpness}} \leq \frac{2}{\eta}.
$$

  Otherwise, GD escapes from that minimum exponentially fast: $(1 - \eta\lambda_1(H(\theta^*)))^t$.
- **Implication:** Stability can control the largest eigenvalue of Hessian.

The Edge of Stability (EoS) Phenomenon

For training neural networks, we find that GD often occurs on the **edge of stability** (EoS)

Table 1: Sharpness $\|H(\theta^*)\|_2$ of GD solutions vs. the learning rate η

η	0.01	0.05	0.1	0.5	1
FashionMNIST	53.5 ± 4.3	39.3 ± 0.5	19.6 ± 0.15	3.9 ± 0.0	1.9 ± 0.0
CIFAR10	198.9 ± 0.6	39.8 ± 0.2	19.8 ± 0.1	3.6 ± 0.4	-
upper bound: $2/\eta$	200	40	20	4	2

See follow-up works (Cohen et al., ICLR 2021; Jastrzebski et al., ICLR 2020) on this striking phenomenon.

GD on Neural Networks Typically Occurs at EoS

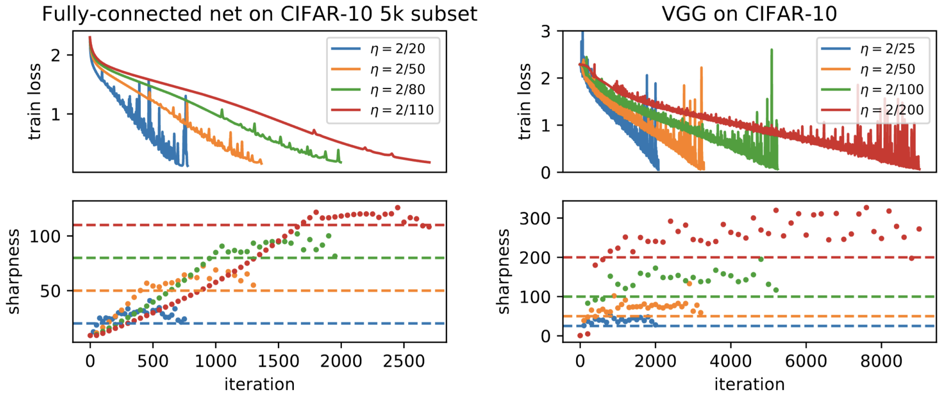


Figure 14: Taken from Cohen et al., (2021).

Remark:

- EoS (Wu et al. (2018)), progressive sharpening (Jastrzebski et al. (2020)).
- Cohen et al., (2021) provides a systematical investigation of the EoS and progressive sharpening phenomenon and highlight the importance of these phenomena.

What affects the stability of SGD

- GD: Consider the optimization of $f(x) = \frac{1}{2}ax^2$, GD will escape the minimum if the learning rate $\eta > 2/a$.

What affects the stability of SGD

- GD: Consider the optimization of $f(x) = \frac{1}{2}ax^2$, GD will escape the minimum if the learning rate $\eta > 2/a$.
- SGD:

$$f_1(x) = \min\left\{\frac{1}{2}x^2, \frac{0.1}{2}(x-1)^2\right\}, \quad f_2(x) = \min\left\{\frac{1}{2}x^2, \frac{1.9}{2}(x-1)^2\right\}$$

  - Both $x = 0$ and $x = 1$ are global minima.
  - The two functions correspond to different batches of data.. GD optimizes $f(x) = \frac{1}{2}(f_1(x) + f_2(x))$.
  - In each iteration, SGD randomly picks one function from f_1 and f_2 and applies gradient descent to that function.
  - SGD with the learning rate $\eta = 0.7$ is not stable around $x = 1$: stable for f_1 but unstable for f_2.

An illustrative example

Consider the target function $f(x) = \frac{1}{2}(f_1(x) + f_2(x))$ with

$$f_1(x) = \min(x^2, 0.1(x-1)^2), \qquad f_2(x) = \min(x^2, 1.9(x-1)^2)$$

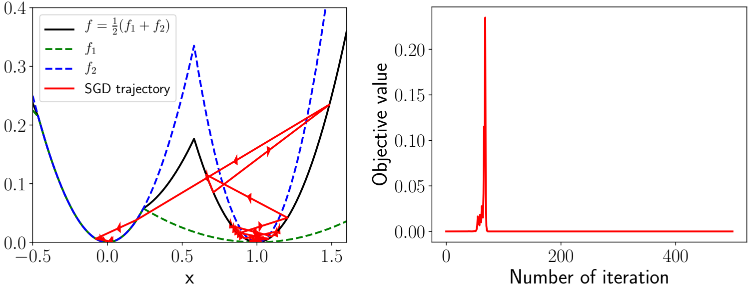


Figure 15: SGD with $\eta = 0.7, x_0 = 1 - \varepsilon$ with ε=1e-5.

Implication:

- Sharpness cannot fully characterize the difference between SGD and GD. The introduction of non-uniformity is necessary.

Linear stability of SGD

- Here we focus on the over-parameterized regime. Then, all global minima are fixed points of SGD since at global minimum:

$$L(\theta^*) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta^*) = 0 \Rightarrow \ell_i(\theta^*) = 0 \Rightarrow \nabla \ell_i(\theta^*) = 0, \forall i = 1, \ldots, n$$

Linear stability of SGD

- Here we focus on the over-parameterized regime. Then, all global minima are fixed points of SGD since at global minimum:

$$L(\theta^*) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta^*) = 0 \Rightarrow \ell_i(\theta^*) = 0 \Rightarrow \nabla \ell_i(\theta^*) = 0, \forall i = 1, \ldots, n$$

- Consider an one-dimensional problem:

$$f(x) = \frac{1}{2n}\sum_{i=1}^{n} a_i x^2, \quad a_i \geq 0 \ \forall i \in [n] \tag{2}$$

The SGD iteration is given by,

$$x_{t+1} = x_t - \eta a_{i_t} x_t = (1 - \eta a_{i_t}) x_t, \tag{3}$$

Linear stability of SGD

- So after one step update, we have

$$\begin{aligned} \mathbb{E}\, x_{t+1} &= (1-\eta a)\, \mathbb{E}\, x_t, && (4) \\ \mathbb{E}\, x_{t+1}^2 &= \left[(1-\eta a)^2 + \eta^2 s^2\right] \mathbb{E}\, x_t^2, && (5) \end{aligned}$$

where $a = \frac{1}{n}\sum_{i=1}^n a_i$, $s = \sqrt{\frac{1}{n}\sum_{i=1}^n a_i^2 - a^2}$. We call a: sharpness s: non-uniformity.

Linear stability of SGD

- So after one step update, we have

$$\begin{aligned}\mathbb{E}\, x_{t+1} &= (1-\eta a)\,\mathbb{E}\, x_t, & (4)\\ \mathbb{E}\, x_{t+1}^2 &= \left[(1-\eta a)^2 + \eta^2 s^2\right]\mathbb{E}\, x_t^2, & (5)\end{aligned}$$

where $a = \frac{1}{n}\sum_{i=1}^n a_i$, $s = \sqrt{\frac{1}{n}\sum_{i=1}^n a_i^2 - a^2}$. We call a: sharpness s: non-uniformity.

- Global minimum $x^* = 0$ is stable for SGD with batch size B, iff

$$(1-\eta a)^2 + \frac{\eta^2(n-B)}{B(n-1)} s^2 \leq 1, \quad s \geq 0. \tag{6}$$

Linear stability of SGD

- So after one step update, we have

$$\begin{aligned} \mathbb{E}\, x_{t+1} &= (1-\eta a)\, \mathbb{E}\, x_t, && (4) \\ \mathbb{E}\, x_{t+1}^2 &= \left[(1-\eta a)^2 + \eta^2 s^2\right] \mathbb{E}\, x_t^2, && (5) \end{aligned}$$

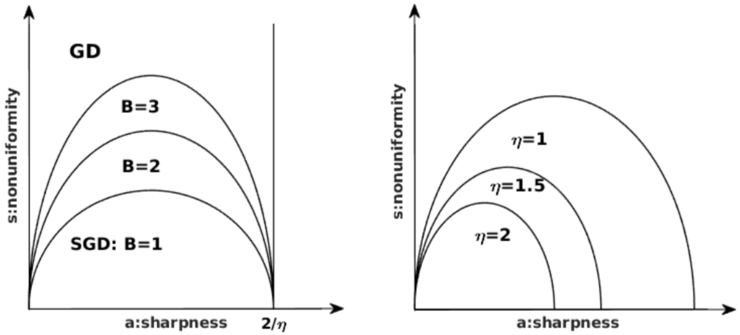
where $a = \frac{1}{n}\sum_{i=1}^n a_i$, $s = \sqrt{\frac{1}{n}\sum_{i=1}^n a_i^2 - a^2}$. We call a: sharpness s: non-uniformity.

- Global minimum $x^* = 0$ is stable for SGD with batch size B, iff

$$(1-\eta a)^2 + \frac{\eta^2(n-B)}{B(n-1)} s^2 \le 1, \quad s \ge 0. \tag{6}$$

- Otherwise, a small perturbation will lead SGD to escape from 0.

The Selection Diagram



The learning rate and batch size play different roles in the global minima selection.

Extension to high dimensions

- Similar analyses can be extended for high-dimensional cases

$$\lambda_{\max}\left\{(I-\eta H)^2+\frac{\eta^2(n-B)}{B(n-1)}\Sigma\right\}\leq 1.$$

Let $a=\lambda_{\max}(H)$, $s^2=\lambda_{\max}(\Sigma)$, then a necessary condition is

$$0\leq a\leq\frac{2}{\eta},\quad 0\leq s\leq\frac{1}{\eta}\sqrt{\frac{B(n-1)}{n-B}}\approx\frac{\sqrt{B}}{\eta}.$$

The selection mechanism

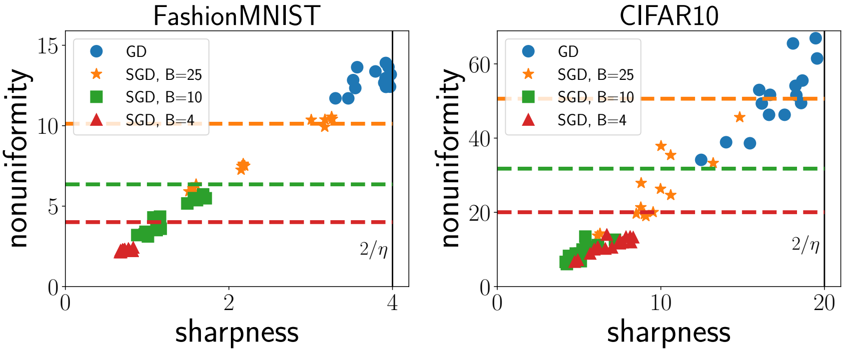


Figure 16: The sharpness-non-uniformity diagram for the minima selected by SGD.

- SGD prefer uniform solutions.
- Non-uniformity is nearly proportional to the sharpness.
- Combining them together, SGD is biased towards flat minima.

Non-uniformity is strongly correlated to sharpness

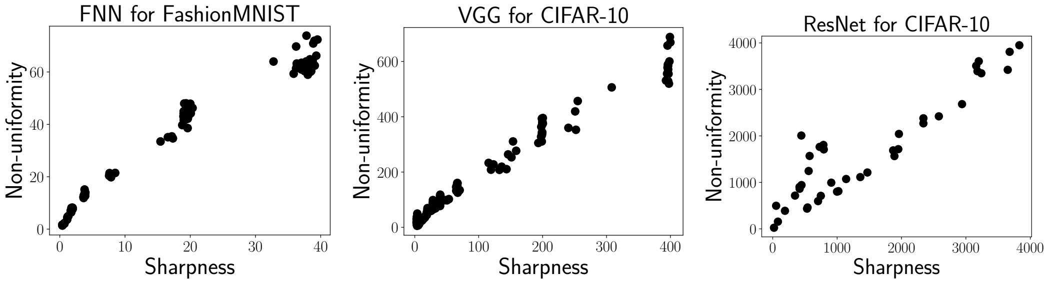


Figure 17: Scatter plot of sharpness and non-uniformity. For each case, we trained about 500 models with different initializations, learning rates, batch sizes, etc.

Towards A Necessary Stability Condition of SGD

- Consider

$$\theta_{t+1} = \theta_t - \eta(\nabla L(x_t) + \xi_t)$$

Towards A Necessary Stability Condition of SGD

- Consider

$$\theta_{t+1} = \theta_t - \eta(\nabla L(x_t) + \xi_t)$$

- Let $\Sigma(\theta_t) = \mathbb{E}[\xi_t \xi_t^\top]$. When $\nabla L(\theta_t)$ or η is small, we have

$$\begin{aligned}\mathbb{E}[L(\theta_{t+1})] &= \mathbb{E}[L(\theta_t - \eta \nabla L(x_t) - \eta \xi_t)] \\ &\approx \mathbb{E}[L(\theta_t - \eta \nabla L(x_t))] + \frac{\eta^2}{B} \mathrm{Tr}[H(\theta_t)\Sigma(\theta_t)].\end{aligned}$$

Towards A Necessary Stability Condition of SGD

- Consider

$$\theta_{t+1} = \theta_t - \eta(\nabla L(x_t) + \xi_t)$$

- Let $\Sigma(\theta_t) = \mathbb{E}[\xi_t \xi_t^\top]$. When $\nabla L(\theta_t)$ or η is small, we have

$$\begin{aligned}\mathbb{E}[L(\theta_{t+1})] &= \mathbb{E}[L(\theta_t - \eta\nabla L(x_t) - \eta\xi_t)] \\ &\approx \mathbb{E}[L(\theta_t - \eta\nabla L(x_t))] + \frac{\eta^2}{B}\mathrm{Tr}[H(\theta_t)\Sigma(\theta_t)].\end{aligned}$$

- The first term comes from the GD part, while the second term is determined by SGD noise.

Towards A Necessary Stability Condition of SGD

- Consider
$$\theta_{t+1} = \theta_t - \eta(\nabla L(x_t) + \xi_t)$$
- Let $\Sigma(\theta_t) = \mathbb{E}[\xi_t \xi_t^\top]$. When $\nabla L(\theta_t)$ or η is small, we have
$$\begin{aligned}\mathbb{E}[L(\theta_{t+1})] &= \mathbb{E}[L(\theta_t - \eta \nabla L(x_t) - \eta \xi_t)] \\ &\approx \mathbb{E}[L(\theta_t - \eta \nabla L(x_t))] + \frac{\eta^2}{B} \mathrm{Tr}[H(\theta_t)\Sigma(\theta_t)].\end{aligned}$$
- The first term comes from the GD part, while the second term is determined by SGD noise.
- Obviously, how SGD noise contributes the stability depends on

how the noise covariance $\Sigma(\theta_t)$ aligns with the Hessian $H(\theta_t)$.

The alignment property of SGD noise

- The decoupling approximation near global minima manifold:

$$
\begin{aligned}
\Sigma(\theta) &= \frac{1}{n}\sum_i e_i \nabla f(\mathbf{x}_i;\theta) e_i \nabla f(\mathbf{x}_i;\theta)^T - \nabla L(\theta)\nabla L(\theta)^T \\
&\approx \frac{1}{n}\sum_i e_i^2 \nabla f(\mathbf{x}_i;\theta)\nabla f(\mathbf{x}_i;\theta)^T \\
&\approx \left(\frac{1}{n}\sum_i e_i^2\right)\left(\frac{1}{n}\sum_i \nabla f(\mathbf{x}_i;\theta)\nabla f(\mathbf{x}_i;\theta)^T\right) = 2L(\theta)G(\theta).
\end{aligned}
$$

The alignment property of SGD noise

- The decoupling approximation near global minima manifold:

$$\begin{aligned}\Sigma(\theta) &= \frac{1}{n}\sum_i e_i \nabla f(\mathbf{x}_i;\theta) e_i \nabla f(\mathbf{x}_i;\theta)^T - \nabla L(\theta)\nabla L(\theta)^T \\ &\approx \frac{1}{n}\sum_i e_i^2 \nabla f(\mathbf{x}_i;\theta)\nabla f(\mathbf{x}_i;\theta)^T \\ &\approx \left(\frac{1}{n}\sum_i e_i^2\right)\left(\frac{1}{n}\sum_i \nabla f(\mathbf{x}_i;\theta)\nabla f(\mathbf{x}_i;\theta)^T\right) = 2L(\theta)G(\theta).\end{aligned}$$

- **Magnitude**: The noise magnitude is proportional to the loss.
- **Direction**: That $\Sigma(\theta)$ aligns with $G(\theta)$ suggests

  Near the global minima manifold, the noise concentrates in sharp directions of local landscape.

Quantify the alignment strength

$$\alpha(\theta) = \frac{\mathrm{Tr}(\Sigma(\theta)G(\theta))}{\|G(\theta)\|_F \|\Sigma(\theta)\|_F} \tag{7}$$

$$\beta(\theta) = \frac{\|\Sigma(\theta)\|_F}{2L(\theta)\|G(\theta)\|_F} \tag{8}$$

$$\mu(\theta) = \alpha(\theta)\beta(\theta) \tag{9}$$

- $\alpha(\theta)$: standard cosine similarity to quantify the "direction" alignment.
- $\beta(\theta)$ quantifies the "magnitude" non-degeneracy of noise wrt the loss.
- $\mu(\theta)$ is a loss-scaled alignment factor.

Quantify the alignment strength

$$\alpha(\theta) = \frac{\mathrm{Tr}(\Sigma(\theta)G(\theta))}{\|G(\theta)\|_F\|\Sigma(\theta)\|_F} \quad (7)$$

$$\beta(\theta) = \frac{\|\Sigma(\theta)\|_F}{2L(\theta)\|G(\theta)\|_F} \quad (8)$$

$$\mu(\theta) = \alpha(\theta)\beta(\theta) \quad (9)$$

- $\alpha(\theta)$: standard cosine similarity to quantify the "direction" alignment.
- $\beta(\theta)$ quantifies the "magnitude" non-degeneracy of noise wrt the loss.
- $\mu(\theta)$ is a loss-scaled alignment factor.

The key observation: There exists a positive constant μ_0 such that

$$\mu(\theta) \geq \mu_0,$$

(When the decoupling approximation holds, $\mu_0 = 1$.)

Experiment results: MNIST

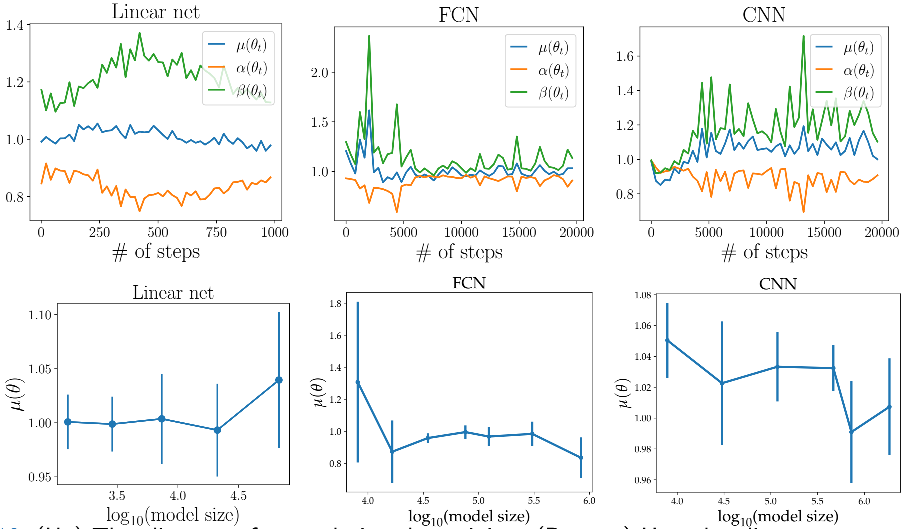
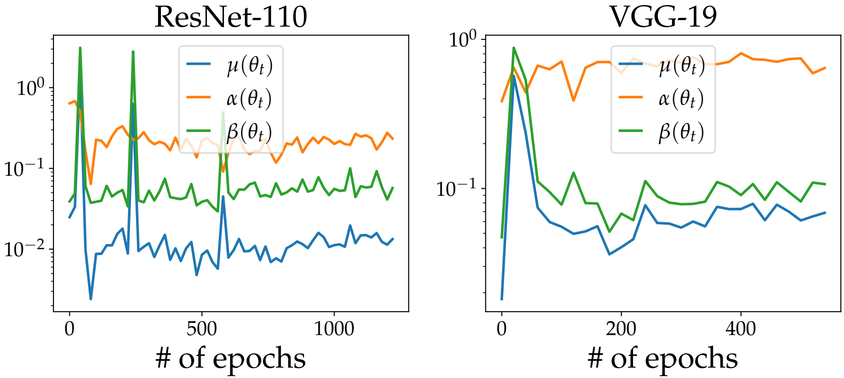


Figure 18: (Up) The alignment factors during the training. (Bottom) How the alignment strength changes with the over-parameterization. Here FCN=fully-connected networks.

Experiment results: CIFAR-10



Why is the alignment satisfied?

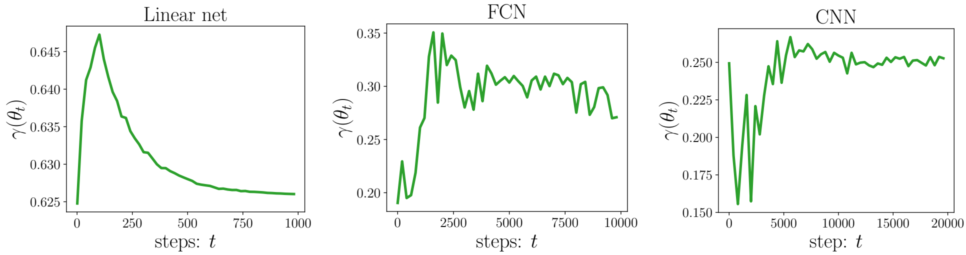
$$\begin{aligned}\mathrm{Tr}(\Sigma(\theta)G(\theta)) &= \frac{1}{n}\sum_{i=1}^{n} e_i^2 g_i(\theta)^T G(\theta) g_i(\theta) = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \|g_i(\theta)\|_G^2 \\ &\approx (\frac{1}{n}\sum_{i=1}^{n} e_i^2)(\frac{1}{n}\sum_{i=1}^{n}\|g_i(\theta)\|_G^2) = 2L(\theta)\|G(\theta)\|_F^2,\end{aligned}$$

the $\approx$ comes from the uniformity of $\{\|g_i(\theta)\|_G\}_i$ are uniform. Let $\gamma(\theta) = \min_i \|g_i(\theta)\|_G^2 / (\frac{1}{n}\sum_{i=1}^n \|g_i\|_G^2)$.

Why is the alignment satisfied?

$$\mathrm{Tr}(\Sigma(\theta)G(\theta)) = \frac{1}{n}\sum_{i=1}^n e_i^2 g_i(\theta)^T G(\theta) g_i(\theta) = \frac{1}{n}\sum_{i=1}^n e_i^2 \|g_i(\theta)\|_G^2$$
$$\approx (\frac{1}{n}\sum_{i=1}^n e_i^2)(\frac{1}{n}\sum_{i=1}^n \|g_i(\theta)\|_G^2) = 2L(\theta)\|G(\theta)\|_F^2,$$

the $\approx$ comes from the uniformity of $\{\|g_i(\theta)\|_G\}_i$ are uniform. Let $\gamma(\theta) = \min_i \|g_i(\theta)\|_G^2/(\frac{1}{n}\sum_{i=1}^n \|g_i\|_G^2)$.



In the literature, many people attribute the validity of approximation to the uniformity of fitting errors $\{e_i^2\}_i$, e.g., (Liu et al., iclr2022), which is unfortunately wrong.

Provable alignments

Proposition 1: Linear networks

Let $f(x;\theta)$ be linear network. Let $f(\cdot;\theta)$ be a deep linear net and $x \sim \mathcal{N}(0, S)$. Consider the online SGD setting, i.e., $n = \infty$. Then, $\mu(\theta) \geq 1$.

Proposition 2: Random feature models

Let $f(x;\theta) = \sum_{j=1}^{m} \theta_j \sigma(w_j^T x)$, where $\{w_j\}_j \stackrel{iid}{\sim} \mathrm{Unif}(\sqrt{d}\mathbb{S}^{d-1})$. Suppose that $x \sim \mathrm{Unif}(\mathbb{S}^{d-1})$. For any $\delta \in (0,1)$, assume $n \gtrsim d^5 \log(1/\delta)$, then w.p. at least $1-\delta$, $\mu(\theta) \gtrsim d^{-1}$.

In these models, we prove that the alignment holds for the entire parameter space not only around global minima.

The linear stability condition

Theorem 3

Let θ^* be a global minimum that is linearly stable. If the noise of linearized SGD satisfies $\mu(\theta) \geq \mu_0$, then

$$\|H(\theta^*)\|_F \leq \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}.$$

The linear stability condition

Theorem 3

Let θ^* be a global minimum that is linearly stable. If the noise of linearized SGD satisfies $\mu(\theta) \geq \mu_0$, then

$$\|H(\theta^*)\|_F \leq \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}.$$

Proof: By the preceding lemma, we have

$$\begin{aligned}
\mathbb{E}[\tilde{L}(\tilde{\theta}_{t+1})] &\geq \frac{\eta^2}{2B}\,\mathbb{E}[\mathrm{Tr}(H(\theta^*)\Sigma(\tilde{\theta}_t))] = \frac{\eta^2\|H(\theta^*)\|_F^2}{B}\,\mathbb{E}[\mu(\theta_t)\tilde{L}(\tilde{\theta}_t)] \\
&\geq \frac{\mu_0\eta^2\|H(\theta^*)\|_F^2}{B}\,\mathbb{E}[\tilde{L}(\tilde{\theta}_t)] \qquad (\text{Using } \mu(\theta) \geq \mu_0).
\end{aligned}$$

The stability ensures $\frac{\mu_0\eta^2\|H(\theta^*)\|_F^2}{B} \leq 1$. Hence, $\|H(\theta^*)\|_F^2 \leq B/(\mu_0\eta^2)$.

Implication: a size-independent flatness control

$$\|H(\theta^*)\|_F \leq \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}.$$

- This upper bound of flatness is independent of the sample and parameter size, no matter how over-parameterized the model is.
- Large LR and small batch size lead to flatter minima.

Implication: a size-independent flatness control

$$\|H(\theta^*)\|_F \leq \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}.$$

- This upper bound of flatness is independent of the sample and parameter size, no matter how over-parameterized the model is.
- Large LR and small batch size lead to flatter minima.
- Comparison with GD.
    - They control different "flatness":

$$\underbrace{\|H(\theta^*)\|_F = \sqrt{\sum_{j=1}^{m} \lambda_j^2(H(\theta^*))} \leq \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}}_{\text{SGD}} \quad \text{vs} \quad \underbrace{\lambda_1(H(\theta^*)) \leq \frac{2}{\eta}}_{\text{GD}}.$$

    - A naive bound of Hessian's Fro-norm for GD:

$$\|H(\theta^*)\|_F \leq \sqrt{\mathrm{rank}(H(\theta^*))}\lambda_1(H(\theta^*)) \leq \frac{2\sqrt{n}}{\eta}.$$

    This is size dependent.

The importance of noise structure

Let m denote the parameter space dimension. Consider two types of SGDs:

$$\begin{aligned} \text{Geometry-aware SGD:} \quad & \theta_{t+1} = \theta_t - \eta(\nabla L(\theta_t) + \xi_{1,t}) \\ \text{Isotropic SGD:} \quad & \theta_{t+1} = \theta_t - \eta(\nabla L(\theta_t) + \xi_{2,t}), \end{aligned}$$

where

$$\mathbb{E}[\xi_{1,t}\xi_{1,t}^T] = 2L(\theta_t)G(\theta_t), \qquad \mathbb{E}[\xi_{2,t}\xi_{2,t}^T] = 2\sigma^2 L(\theta_t) I_m,$$

where $\sigma^2 = \frac{\mathrm{Tr}(G(\theta_t))}{m}$ is chosen to ensure that two types of noises have the same total variance.

The importance of noise structure

Let m denote the parameter space dimension. Consider two types of SGDs:

$$\begin{aligned} \text{Geometry-aware SGD:} \quad & \theta_{t+1} = \theta_t - \eta(\nabla L(\theta_t) + \xi_{1,t}) \\ \text{Isotropic SGD:} \quad & \theta_{t+1} = \theta_t - \eta(\nabla L(\theta_t) + \xi_{2,t}), \end{aligned}$$

where

$$\mathbb{E}[\xi_{1,t}\xi_{1,t}^T] = 2L(\theta_t)G(\theta_t), \qquad \mathbb{E}[\xi_{2,t}\xi_{2,t}^T] = 2\sigma^2 L(\theta_t) I_m,$$

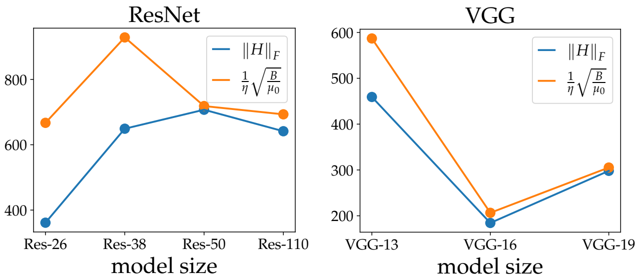
where $\sigma^2 = \frac{\mathrm{Tr}(G(\theta_t))}{m}$ is chosen to ensure that two types of noises have the same total variance.

The stability of two SGDs:

$$\text{Geometry-aware SGD:} \qquad \|H(\theta^*)\|_F \leq \frac{\sqrt{B}}{\eta} \quad (\text{size-independent})$$

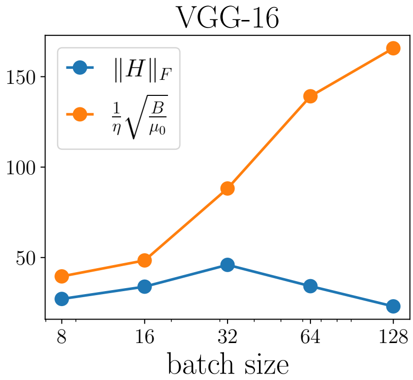
$$\text{Isotropic SGD:} \qquad \mathrm{Tr}(H(\theta^*)) \leq \frac{\sqrt{mB}}{\eta} \quad (\text{size-dependent}).$$

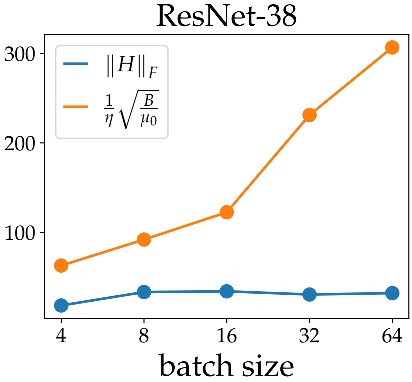
CIFAR-10 experiments



- The actual sharpness of SGD solutions is (nearly) independent of the model size.
- Our upper bound is close to the actual sharpness, suggesting a **near EoS phenomenon** for SGD.

The bound becomes tighter as decreasing batch size





How much SGD dislikes sharp minima?

Theorem 4 (Escape from sharp minima)

If $\|H(\theta^*)\|_F > \frac{1}{\eta}\sqrt{\frac{B}{\mu_0}}$*, then we have*

$$\mathbb{E}[\hat{L}(\theta_t)] \geq \gamma_0^t\, \mathbb{E}[\hat{L}(\theta_0)]$$

where $\gamma_0 = \frac{\eta^2 \mu_0}{B}\|H(\theta^*)\|_F^2 > 1$.

- The sharper the minimum is, the faster the escape is.
- The stronger the noise aligns with local geometry, the faster the escape is.

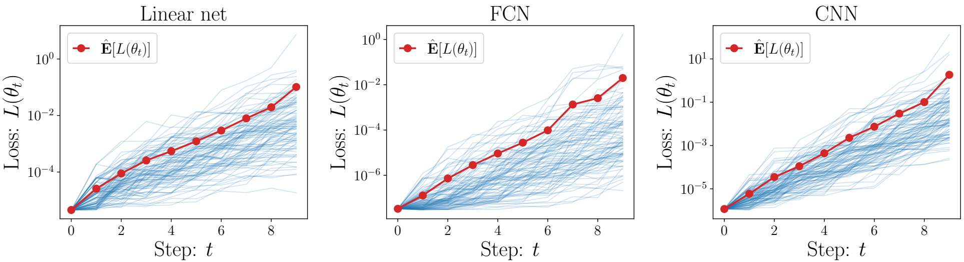


Figure 19: **The exponentially fast escape from sharp minima.** The blue curves are 200 trajectories of SGD; the red curve corresponds to the average. The sharp minimum is found by GD. When GD nearly converge, we switch to SGD with the same learning rate. This choice ensures that the minimum is stable for GD, and thus the escape is purely driven by SGD noise.

References I

[1] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, “Visualizing the loss landscape of neural nets,” in *Advances in Neural Information Processing Systems*, 2018, pp. 6389–6399.

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

[3] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[4] P. Alquier, “User-friendly introduction to PAC-Bayes bounds,” *arXiv preprint arXiv:2110.11216*, 2021.

[5] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” in *International Conference on Learning Representations*, 2020.

[6] L. Wu and W. J. Su, “The implicit regularization of dynamical stability in stochastic gradient descent,” in *International Conference on Machine Learning*, PMLR, 2023, pp. 37 656–37 684.

References II

[7] K. Wen, Z. Li, and T. Ma, "Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.