目的関数の平滑化とディープニューラルネットワークの汎化性能におけるモーメンタム法の慣性項の役割

日本オペレーションズ・リサーチ学会 2024年秋季研究発表会
9/11(水) 10:10～10:30

明治大学　佐藤尚樹
明治大学　飯塚秀明

背景：関数の平滑化

関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ を、大きさ $\delta \in \mathbb{R}$ のノイズで平滑化した関数は、

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u} \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{\sqrt{d}} I_d\right)} \left[f(\boldsymbol{x} - \delta \boldsymbol{u}) \right]$$

確率変数 $\boldsymbol{u} \in \mathbb{R}^d$

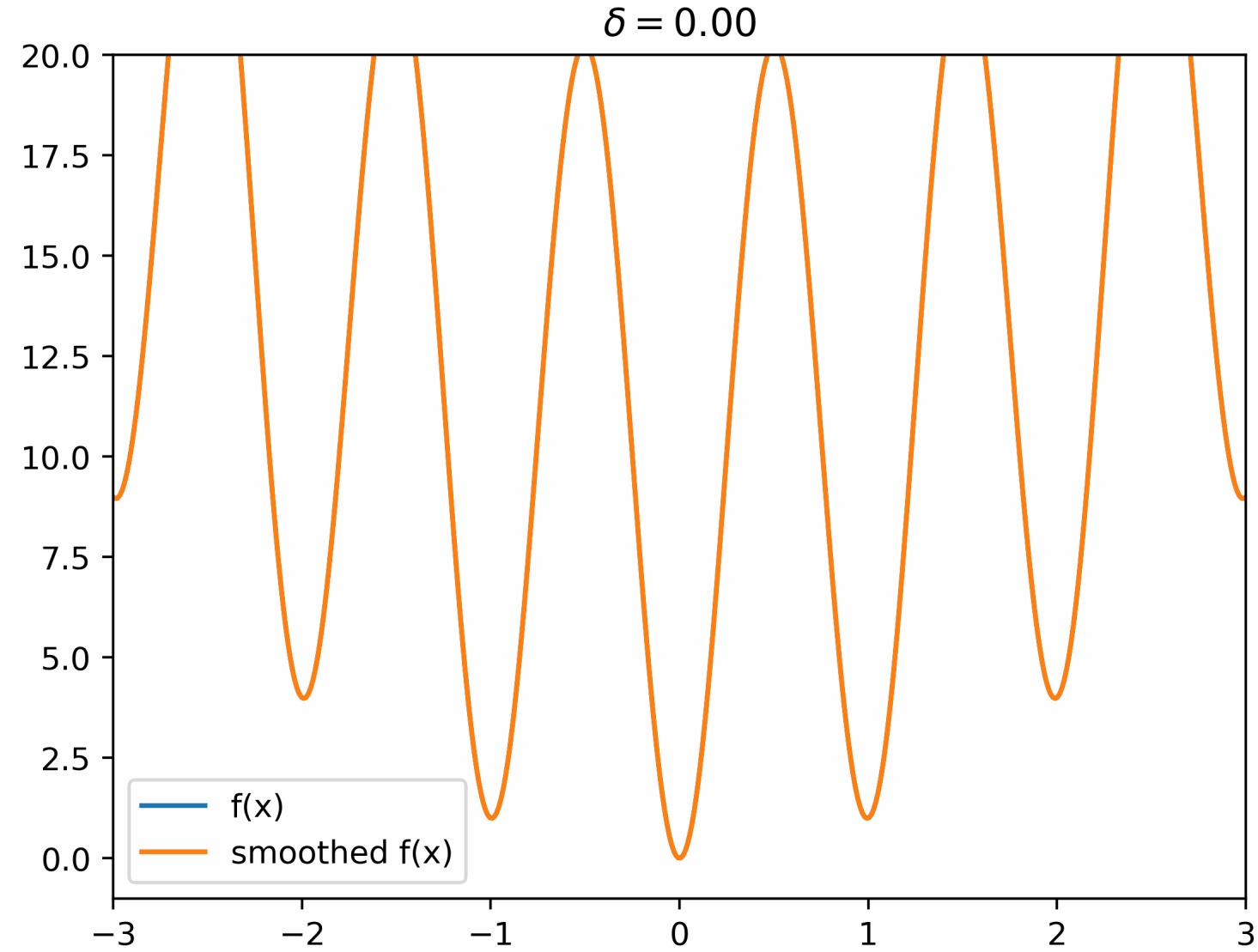
平均 $\mathbf{0}$, 分散 $\frac{1}{\sqrt{d}} I_d$ の正規分布

ノイズスケール（平滑化の度合い）

背景：関数の平滑化

▷**目的関数の平滑化**

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}\left(\mathbf{0},\frac{1}{\sqrt{d}}I_d\right)}[f(\boldsymbol{x}-\delta\boldsymbol{u})]$$



▷1変数のRastrigin's 関数

$$f(x) = x^2 - 10\cos(2\pi x) + 10$$

背景：機械学習（画像分類）

入力

$\boldsymbol{z}_i \in \mathbb{R}^d$

$(i = 1, 2, \cdots, N)$

$d = 512 \times 512 \times 3$

$N = 3000000$

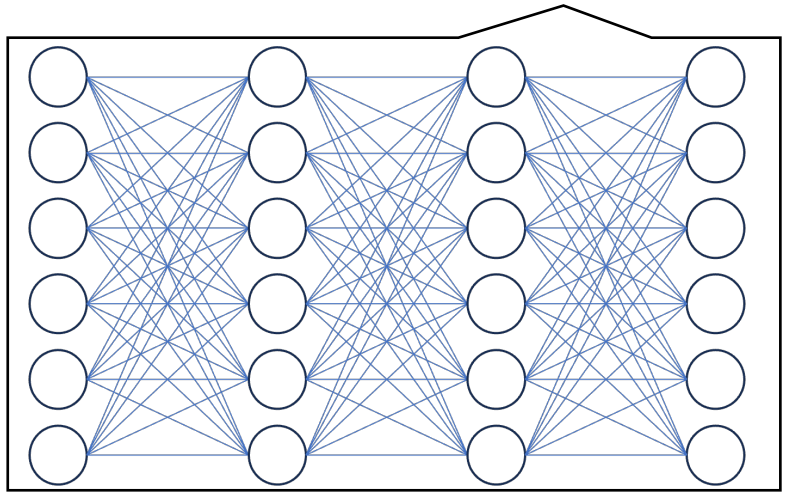
データセット

関数 $g\colon \mathbb{R}^d \to \mathbb{R}$

変数: $\boldsymbol{x} \in \mathbb{R}^D$,

$\boldsymbol{z}_i \in \mathbb{R}^d$

Deep Neural Network



$D = 20$**万～**5**兆**

出力

$g(\boldsymbol{x}, \boldsymbol{z}_i) \in \mathbb{R}$

犬:3

正解

$y_i \in \mathbb{R}$

猫:2

誤差 $|g(\boldsymbol{x}, \boldsymbol{z}_i) - y_i|$

誤差の平均

$$f(\boldsymbol{x}) := \frac{1}{N} \sum_{i=1}^{N} |g(\boldsymbol{x}, \boldsymbol{z}_i) - y_i|$$

関数 $f(\boldsymbol{x})$**を最適化する**

背景：連続最適化

▷**最急降下法** (Gradient Descent)

$$\boldsymbol{d}_t := -\nabla f(\boldsymbol{x}_t)$$

全勾配

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t + \eta_t \boldsymbol{d}_t$$

▷**確率的勾配降下法** (Stochastic Gradient Descent, SGD)

$$\boldsymbol{d}_t := -\nabla f_{S_t}(\boldsymbol{x}_t) = -\frac{1}{b}\sum_{i=1}^{b} \mathsf{G}_{\xi_{t,i}}(\boldsymbol{x}_t)$$

ミニバッチ確率的勾配

確率的勾配

バッチサイズ

ランダムに選ばれたb個の確率的勾配の平均で代用(ex. b=32, 1024)

背景：連続最適化

▷**確率的勾配降下法** (Stochastic Gradient Descent, SGD)

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta_t \nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)$$

▷**慣性項付き確率的勾配降下法** (SGD with momentum)

$$\boldsymbol{m}_t := \nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) + \beta \boldsymbol{m}_{t-1}$$

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta_t \boldsymbol{m}_t$$

慣性係数

ただし、$\beta \in [0, 1), \boldsymbol{m}_{-1} = \boldsymbol{0}$ とする。

▷SGDよりも収束が速く、しかも**汎化性能**も優れる。

背景：機械学習における汎化能力

▷学習が完了したモデルが、訓練データ以外の入力に対しても正しく分類できることを『汎化性に優れる』という。

訓練に使われたデータ

訓練に使われていないデータ

関数 $g: \mathbb{R}^d \to \mathbb{R}$

変数: $\boldsymbol{x} \in \mathbb{R}^D$,

$\boldsymbol{z}_i \in \mathbb{R}^d$

Deep Neural Network

訓練済み

＝パラメータ調整済

猫

経験損失: 0.001

訓練精度: 99.9%

これも猫！

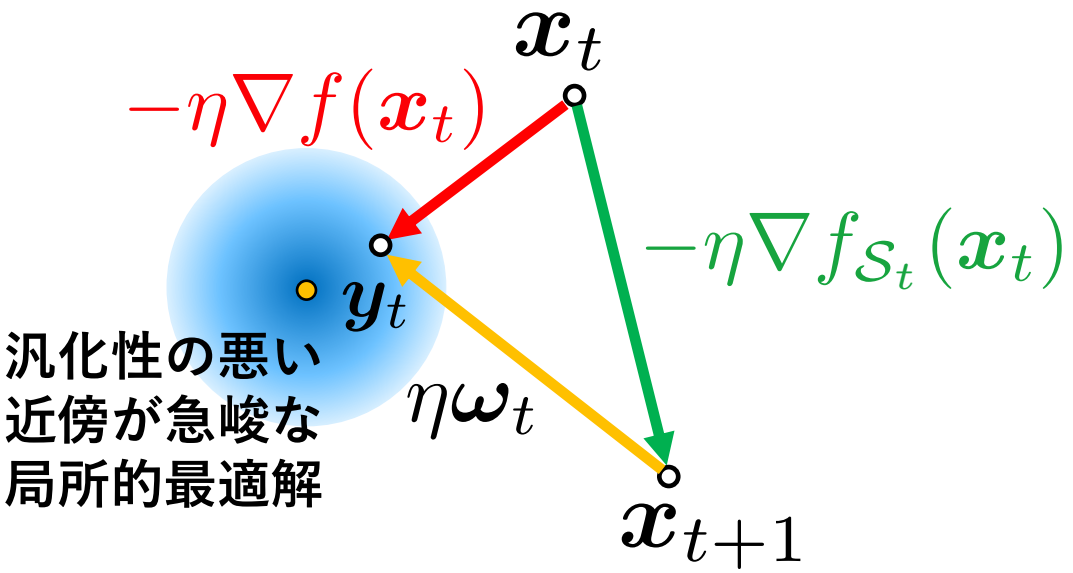
テスト精度: 75%

↑これが汎化性能！

背景：SGDの確率的ノイズと平滑化

▷**非凸関数の最適化において，確率的勾配降下法(SGD)は、なぜか最急降下法(GD)よりも汎化性の高い解に収束する。**

▷**SGDの確率的ノイズが役立っている？[1]**



$$(\mathrm{GD})\ \boldsymbol{y}_t := \boldsymbol{x}_t - \eta\nabla f(\boldsymbol{x}_t)$$

$$(\mathrm{SGD})\ \boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta\nabla f_{S_t}(\boldsymbol{x}_t)$$

SGDの確率的ノイズ

$$\boldsymbol{\omega}_t^{\mathrm{SGD}} := \underbrace{\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)}_{\text{SGDの探索方向}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\text{GDの探索方向（最急降下方向）}}$$

[1] R. Kleinberg et al. “An Alternative View: When Does SGD Escape Local Minima?” In *Proceedings of the 35th International Conference on Machine Learning*, pp2703-2712, 2018

背景：SGDの確率的ノイズと平滑化

$$(\mathrm{GD})\ \boldsymbol{y}_t := \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t)$$

$$(\mathrm{SGD})\ \boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta \nabla f_{S_t}(\boldsymbol{x}_t)$$

▷**目的関数の平滑化**

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u} \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{\sqrt{d}} I_d\right)} [f(\boldsymbol{x} - \delta \boldsymbol{u})]$$

(確率的ノイズ) $\boldsymbol{\omega}_t^{\mathrm{SGD}} := \nabla f_{S_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)$ とすると、次の式が成り立つ。

$$\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[\boldsymbol{y}_{t+1}] = \mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[\boldsymbol{y}_t] - \eta \underbrace{\nabla \mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[f(\boldsymbol{y}_t - \eta \boldsymbol{\omega}_t)]}_{=: \hat{f}(\boldsymbol{y}_t)}$$

▷**“関数$f(x_t)$をSGDで最適化すること”と、“関数$\hat{f}(y_t)$を最急降下法で最適化すること”は、期待値の意味では等価であると言える。**

▷**ある程度平滑化された関数が、最急降下法で最適化されているとみなせる。**

[1] R. Kleinberg et al. “An Alternative View: When Does SGD Escape Local Minima?” In *Proceedings of the 35th International Conference on Machine Learning*, pp2703-2712, 2018

背景：SGDの平滑化特性

[2] N.Sato and H. Iiduka “Using Stochastic Gradient Descent to Smooth Nonconvex Functions: Analysis of Implicit Graduated Optimization with Optimal Noise Scheduling”, arXiv2311.08745 (2023)

補題A.1

$$\mathbb{E}_{\boldsymbol{\xi}_t}\left[\left\|\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)-\nabla f(\boldsymbol{x}_t)\right\|^2\right]\leq\frac{C_{\mathrm{opt}}^2}{b}$$

▷**目的関数の平滑化**

$$\hat{f}_\delta(\boldsymbol{x}):=\mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}\left(\mathbf{0},\frac{1}{\sqrt{d}}I_d\right)}\left[f(\boldsymbol{x}-\delta\boldsymbol{u})\right]$$

▷ $\boldsymbol{\omega}_t^{\mathrm{SGD}}:=\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)-\nabla f(\boldsymbol{x}_t)$ **と補題A.1から、**

$$\boldsymbol{\omega}_t^{\mathrm{SGD}}=\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}}\boldsymbol{u}_t\quad\left(\boldsymbol{u}_t\sim\mathcal{N}\left(\mathbf{0};\frac{1}{\sqrt{d}}I_d\right)\right)$$

が成り立つ。したがって、

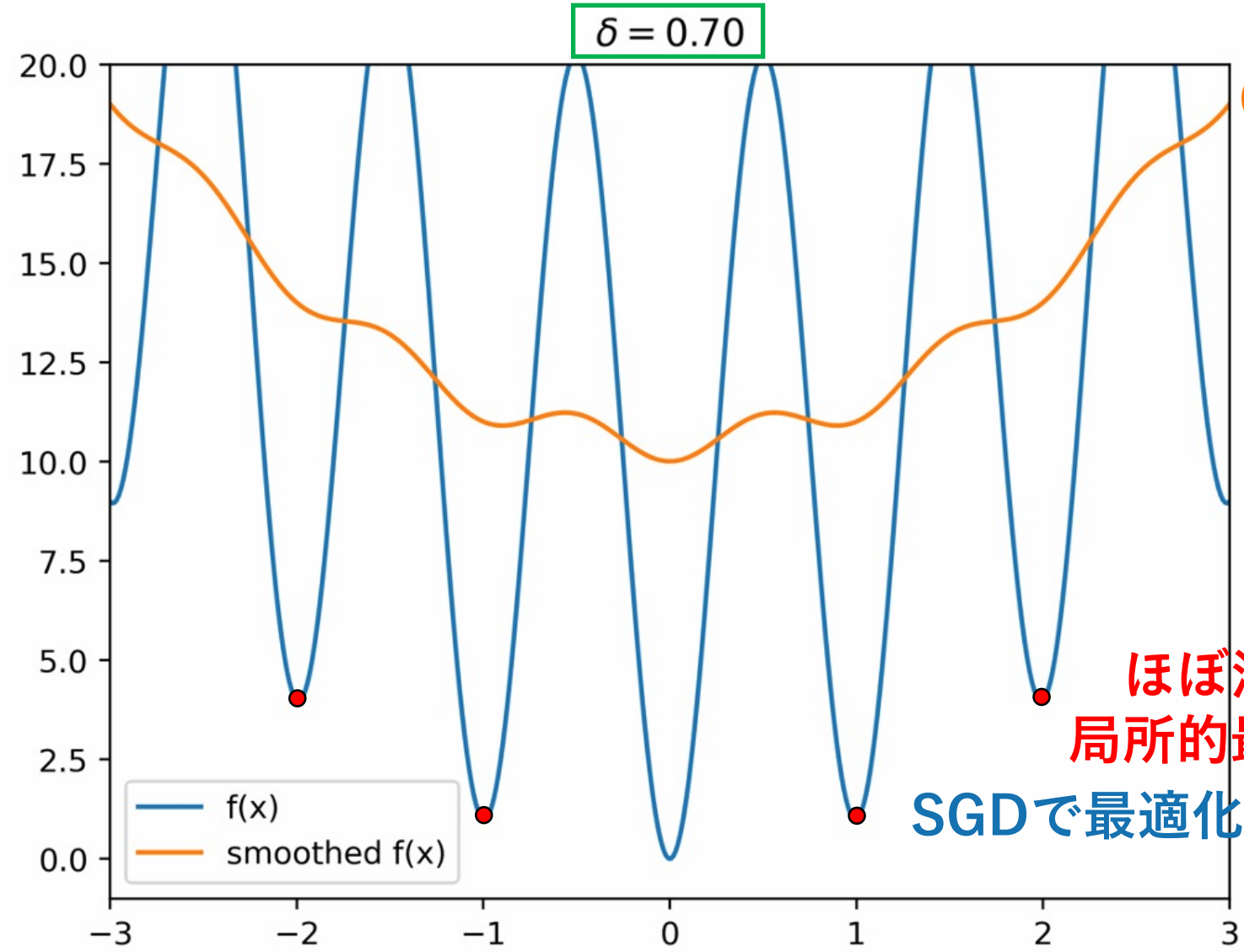
$$\begin{aligned}\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[\boldsymbol{y}_{t+1}]&=\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[\boldsymbol{y}_t]-\eta\nabla\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[f(\boldsymbol{y}_t-\eta\boldsymbol{\omega}_t^{\mathrm{SGD}})]\\&=\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[\boldsymbol{y}_t]-\eta\nabla\mathbb{E}_{\boldsymbol{u}_t\sim\mathcal{N}\left(\mathbf{0};\frac{1}{\sqrt{d}}I_d\right)}\left[f\left(\boldsymbol{y}_t-\eta\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}}\boldsymbol{u}_t\right)\right]\\&=\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SGD}}}[\boldsymbol{y}_t]-\eta\nabla\hat{f}_{\eta\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}}}(\boldsymbol{y}_t)\end{aligned}$$

が成り立つ。

大きさ $\eta C_{\mathrm{SGD}}/\sqrt{b}$ のノイズで平滑化された関数

$$\delta^{\mathrm{SGD}}=\eta\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}}$$

背景：SGDの確率的ノイズによる暗黙的な目的関数の平滑化



GDで最適化されると
みなせる目的関数

$$\delta^{\mathrm{SGD}} = \eta\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}}$$

確率的ノイズの大きさ
＝平滑化の度合い

ほぼ消滅した
局所的最適解たち

SGDで最適化される目的関数

背景：慣性項と確率的ノイズの矛盾

▷**慣性項は確率的ノイズを削減しているはず。**

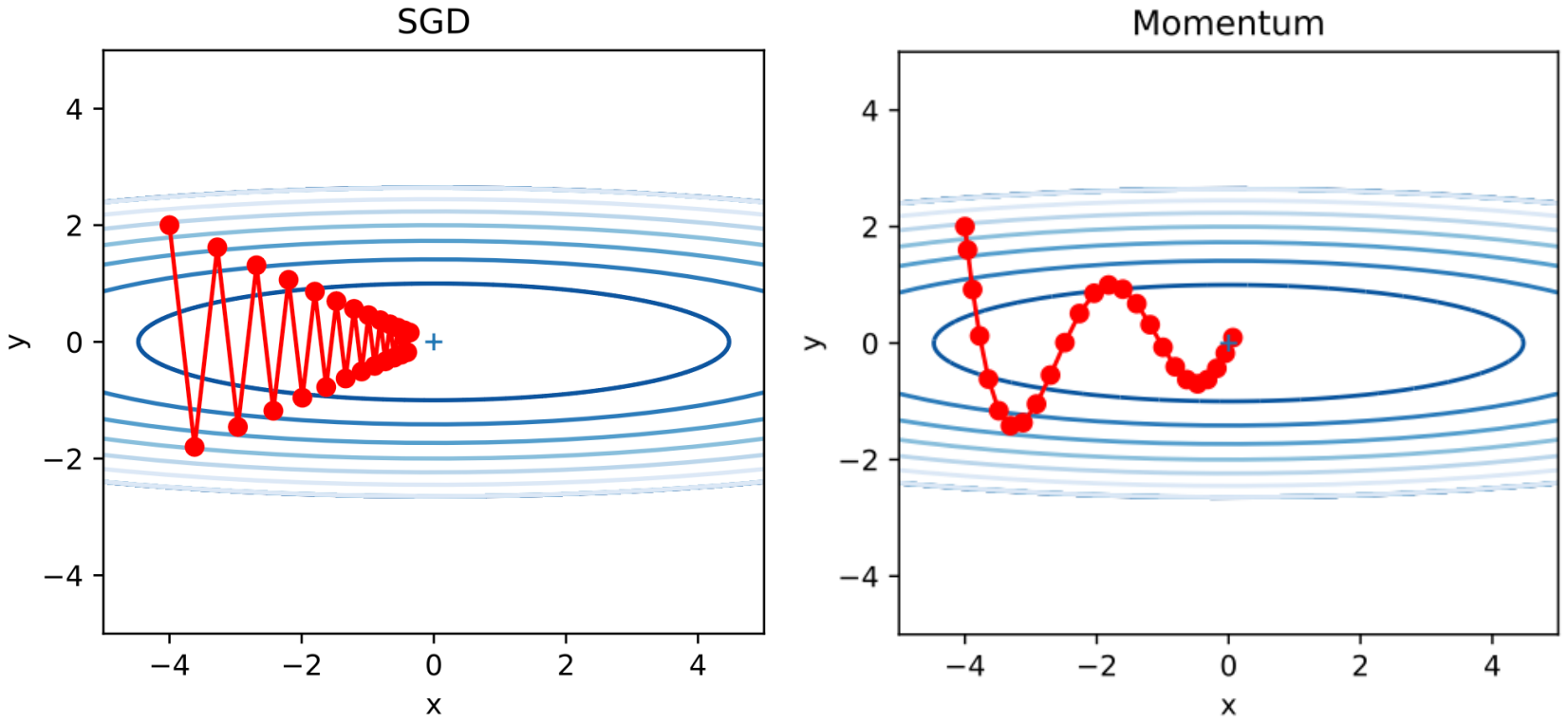
▷SGD

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta_t \nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)$$

▷Momentum

$$\boldsymbol{m}_t := \nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) + \beta \boldsymbol{m}_{t-1}$$

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta_t \boldsymbol{m}_t$$



▷**十分な大きさの確率的ノイズが高い汎化性をもたらすはず。**

▷**しかし、Momentumの方がSGDよりも高い汎化性をもたらす。**

動機

▷慣性項と確率的ノイズの間にある矛盾を解決したい。

＜方針＞

▷SGDの確率的ノイズによる目的関数の平滑化の議論を
SGD with momentumに拡張する。

▷慣性係数 β が平滑化の度合いにどのように現れるかを確認する。

▷数値実験で理論を検証する。

貢献

▷SGD with momentumの確率的ノイズによる平滑化の度合いは、

$$\delta^{\mathrm{SHB}} = \eta\sqrt{\left(1+\hat{\beta}\right)\frac{C^2_{\mathrm{SHB}}}{b} + \hat{\beta}K^2_{\mathrm{SHB}}},\ \delta^{\mathrm{NSHB}} = \eta\sqrt{\frac{1}{1-\beta}\frac{C^2_{\mathrm{NSHB}}}{b}}$$

で表せることを示した。ただし、ηは学習率、bはバッチサイズ、βは慣性係数、C^2は確率的勾配の分散とする。

▷なぜ、いつ、慣性項が汎化性能を向上させるか明らかにした。

▷慣性項と確率的ノイズの間にある矛盾を解決した。

すなわち、「慣性項によって削減される確率的ノイズ」と、「汎化性能に寄与する確率的ノイズ」が別物であることを示した。

準備：最適化問題

経験損失最小化問題

▷訓練データセット $S := (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n)$

▷Deep Neural Network のパラメータ $\boldsymbol{x} \in \mathbb{R}^d$

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}; S) = \frac{1}{n}\sum_{i=1}^{n} l(\boldsymbol{x}; \boldsymbol{z}_i) = \frac{1}{n}\sum_{i=1}^{n} l_i(\boldsymbol{x})$$

i 番目の訓練データ $\boldsymbol{z}_i$ に対する損失関数

▷非凸目的関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ を最適化する。

準備：仮定

仮定

(A1)関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ **は連続的微分可能で、**L_f **-リプシッツ関数とする。**

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d\colon |f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L_f \|\boldsymbol{x} - \boldsymbol{y}\|$$

(A2) $(\boldsymbol{x}_t)_{t\in\mathbb{N}} \subset \mathbb{R}^d$ **を最適化手法によって生成された点列とするとき、**

(i)任意の $t \in \mathbb{N}$ **に対して、次の式が成り立つとする。**

$$\mathbb{E}_{\xi_t}\left[\mathsf{G}_{\xi_t}(\boldsymbol{x}_t)\right] = \nabla f(\boldsymbol{x}_t)$$

(ii)次の式を満たす非負定数 C_{opt}^2 **が存在するとする。**

$$\mathbb{E}_{\xi_t}\left[\|\mathsf{G}_{\xi_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\|^2\right] \leq C_{\mathrm{opt}}^2$$

←各最適化手法の確率的勾配の分散

最適化手法に依存する

準備：仮定

仮定続き

(A3) 時刻$t \in \mathbb{N}$で、全勾配∇fはミニバッチ$\mathcal{S}_t$で次のように近似されるとする。

$$\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) := \frac{1}{b}\sum_{i\in[b]} \mathsf{G}_{\xi_{t,i}}(\boldsymbol{x}_t) = \frac{1}{b}\sum_{\{i:\, \boldsymbol{z}_i \in \mathsf{S}_t\}} \nabla l_i(\boldsymbol{x}_t)$$

ミニバッチ確率的勾配

確率的勾配

(A4) 任意の$t \in \mathbb{N}$で、それぞれの最適化手法に対して、次の式を満たす正の定数K_{opt}が存在するとする。

$$\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}_t)\right\|^2\right] \leq K_{\mathrm{opt}}^2$$

準備：アルゴリズム

[3] I. Gitman, H. Lang, P. Zhang, and L. Xiao, "Understanding the role of momentum in stochastic gradient methods," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 9630–9640.

Algorithm 1 Stochastic Heavy Ball (SHB)

Require: $\boldsymbol{x}_0 \in \mathbb{R}^d, \eta > 0, \beta \in [0,1), \boldsymbol{m}_{-1} := \boldsymbol{0}$
  for $t = 0$ to $T-1$ **do**
    $\boldsymbol{m}_t := \nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) + \beta \boldsymbol{m}_{t-1}$
    $\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta \boldsymbol{m}_t$
  end for
  return $\boldsymbol{x}_T$

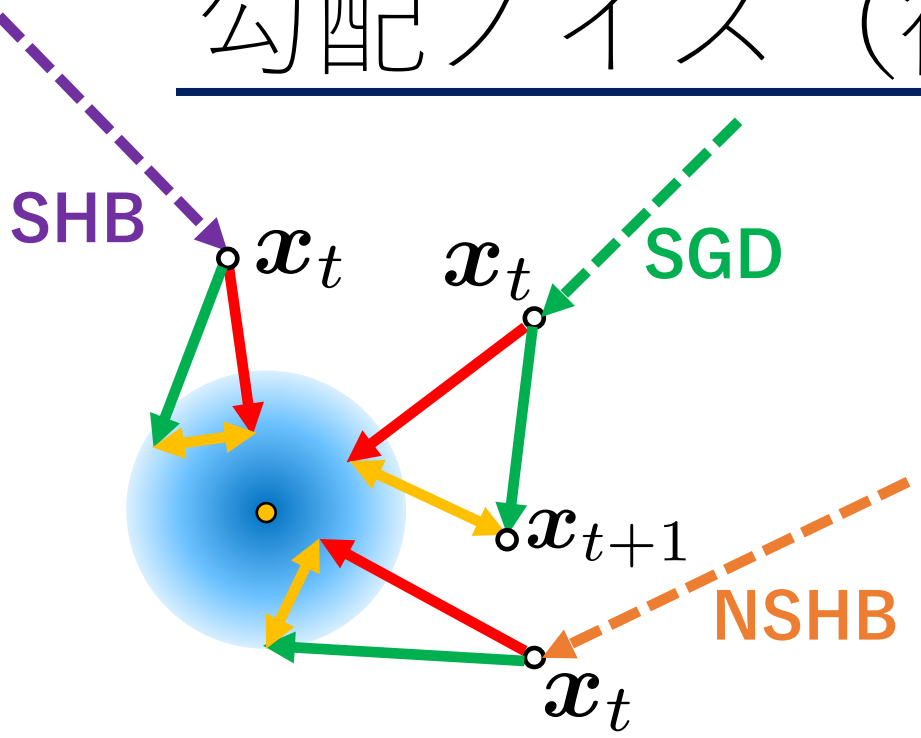
▷凸結合をとらない
▷あまり理論解析されない
▷数値実験でよく利用される
  ▷PyTorchやTensorFlowが提供するSGD with momentum

Algorithm 2 Normalized Stochastic Heavy Ball (NSHB)

Require: $\boldsymbol{x}_0 \in \mathbb{R}^d, \eta > 0, \beta \in [0,1), \boldsymbol{d}_{-1} := \boldsymbol{0}$
  for $t = 0$ to $T-1$ **do**
    $\boldsymbol{d}_t := (1-\beta)\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) + \beta \boldsymbol{d}_{t-1}$
    $\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta \boldsymbol{d}_t$
  end for
  return $\boldsymbol{x}_T$

▷凸結合をとる
▷よく理論解析される
▷数値実験で利用されない

勾配ノイズ（従来の確率的ノイズ）



SGDの勾配ノイズ $\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)$

SHBの勾配ノイズ $\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)$

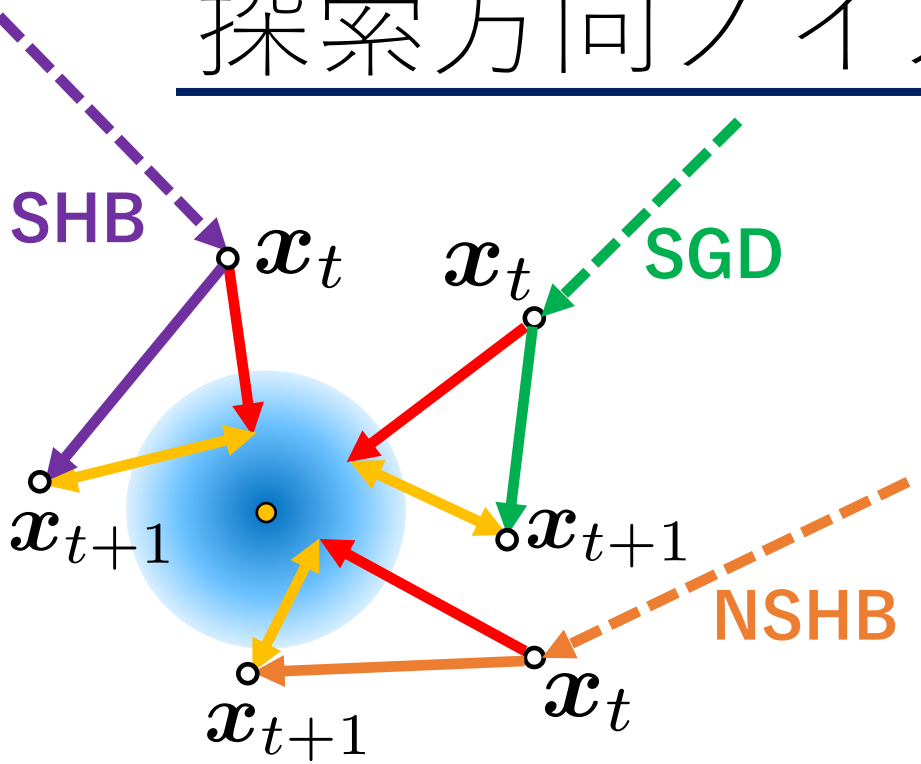
NSHBの勾配ノイズ $\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)$

補題A.1. 仮定(A2)(ii)と(A3)の下で、任意の時刻 $t \in \mathbb{N}$ に対して、次が成り立つ。

$$\mathbb{E}_{\boldsymbol{\xi}_t}\left[\left\|\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2\right] \leq \frac{C_{\mathrm{opt}}^2}{b}$$

∴勾配ノイズ＝確率的勾配の分散

探索方向ノイズ



SGDの探索方向ノイズ $\boldsymbol{\omega}_t^{\mathrm{SGD}} := \underbrace{\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)}_{\text{SGDの探索方向}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\text{GDの探索方向}}$

SHBの探索方向ノイズ $\boldsymbol{\omega}_t^{\mathrm{SHB}} := \underbrace{\boldsymbol{m}_t}_{\substack{\text{SHB} \\ \text{の探索方向}}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\text{GDの探索方向}}$

NSHBの探索方向ノイズ $\boldsymbol{\omega}_t^{\mathrm{NSHB}} := \underbrace{\boldsymbol{d}_t}_{\substack{\text{NSHB} \\ \text{の探索方向}}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\text{GDの探索方向}}$

定理3.1. 仮定(A2)(ii),(A3),(A4)の下で、任意の時刻 $t \in \mathbb{N}$ に対して、次が成り立つ。

$$\mathbb{E}\left[\|\boldsymbol{\omega}_t^{\mathrm{SHB}}\|\right] \leq \sqrt{\frac{C_{\mathrm{SHB}}^2}{b} + \frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}\left(\frac{C_{\mathrm{SHB}}^2}{b} + K_{\mathrm{SHB}}^2\right)}, \quad \mathbb{E}\left[\|\boldsymbol{\omega}_t^{\mathrm{NSHB}}\|\right] \leq \sqrt{\frac{1}{1-\beta}\frac{C_{\mathrm{NSHB}}^2}{b}}$$

SHBの探索方向ノイズと平滑化

$$(\mathrm{GD})\ \boldsymbol{y}_t := \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t)$$

$$(\mathrm{SHB})\ \boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta \boldsymbol{m}_t$$

▷目的関数の平滑化

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{0}, \frac{1}{\sqrt{d}} I_d\right)} [f(\boldsymbol{x} - \delta \boldsymbol{u})]$$

探索方向ノイズ $\boldsymbol{\omega}_t^{\mathrm{SHB}} := \boldsymbol{m}_t - \nabla f(\boldsymbol{x}_t)$ **とすると、次の式が成り立つ。**

$$\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}[\boldsymbol{y}_{t+1}] = \mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}[\boldsymbol{y}_t] - \eta \nabla \underbrace{\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}[f(\boldsymbol{y}_t - \eta \boldsymbol{\omega}_t)]}_{=: \hat{f}(\boldsymbol{y}_t)}$$

▷**“関数** $f(x_t)$**をSHBで最適化すること”と、“関数** $\hat{f}(y_t)$**を最急降下法で最適化すること”は、期待値の意味では等価であると言える。**

▷**ある程度平滑化された関数が、最急降下法で最適化されているとみなせる。**

SHBの平滑化特性

▷目的関数の平滑化

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}\left(\mathbf{0},\frac{1}{\sqrt{d}}I_d\right)}\left[f(\boldsymbol{x}-\delta\boldsymbol{u})\right]$$

定理3.1

$$\mathbb{E}\left[\left\|\boldsymbol{\omega}_t^{\mathrm{SHB}}\right\|\right] \leq \sqrt{\left(1+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}\right)\frac{C_{\mathrm{SHB}}^2}{b}+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}K_{\mathrm{SHB}}^2} =: A$$

▷定理3.1から、

$$\boldsymbol{\omega}_t^{\mathrm{SHB}} = \sqrt{\left(1+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}\right)\frac{C_{\mathrm{SHB}}^2}{b}+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}K_{\mathrm{SHB}}^2}\boldsymbol{u_t} = A\boldsymbol{u}_t$$

$$\left(\boldsymbol{u}_t\sim\mathcal{N}\left(\mathbf{0};\frac{1}{\sqrt{d}}I_d\right)\right)$$

が成り立つ。したがって、

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}\left[\boldsymbol{y}_{t+1}\right] &= \mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}\left[\boldsymbol{y}_t\right]-\eta\nabla\mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}\left[f(\boldsymbol{y}_t-\eta\boldsymbol{\omega}_t^{\mathrm{SHB}})\right]\\ &= \mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}\left[\boldsymbol{y}_t\right]-\eta\nabla\mathbb{E}_{\boldsymbol{u}_t\sim\mathcal{N}\left(\mathbf{0};\frac{1}{\sqrt{d}}I_d\right)}\left[f(\boldsymbol{y}_t-\eta A\boldsymbol{u}_t)\right]\\ &= \mathbb{E}_{\boldsymbol{\omega}_t^{\mathrm{SHB}}}\left[\boldsymbol{y}_t\right]-\eta\nabla\hat{f}_{\eta A}(\boldsymbol{y}_t)\end{aligned}$$

大きさηAのノイズで
平滑化された関数

が成り立つ。

最適化手法ごとの平滑化の度合い

$$\delta^{\mathrm{SGD}} = \eta\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}},$$

∴探索方向ノイズ＝平滑化の度合い

$$\delta^{\mathrm{SHB}} = \eta\sqrt{\left(1+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}\right)\frac{C_{\mathrm{SHB}}^2}{b}+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}K_{\mathrm{SHB}}^2},$$

≈ 3.13

$$\delta^{\mathrm{NSHB}} = \eta\sqrt{\frac{1}{1-\beta}\frac{C_{\mathrm{NSHB}}^2}{b}}$$

▷どれくらいの大きさか？

▷未知数の推定が必要

仮定(A2)(ii) [確率的勾配の分散]

$$\mathbb{E}_{\xi_t}\left[\left\|\mathsf{G}_{\xi_t}(\boldsymbol{x}_t)-\nabla f(\boldsymbol{x}_t)\right\|^2\right] \le C_{\mathrm{opt}}^2$$

仮定(A4) [勾配ノルムの上界]

$$\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}_t)\right\|^2\right] \le K_{\mathrm{opt}}^2$$

確率的勾配の分散 C_{opt}^2 の推定

命題4.1. 仮定(A1)-(A4)の下で、任意の $\epsilon > 0$ に対して、次が成り立つ。

$$b_{\mathrm{SGD}}^\star > \frac{\eta C_{\mathrm{SGD}}^2}{\epsilon^2},\ b_{\mathrm{SHB}}^\star > \frac{\eta(\beta^2-\beta+1)C_{\mathrm{SHB}}^2}{\beta(1-\beta)^2\epsilon^2},\ b_{\mathrm{NSHB}}^\star > \frac{\eta C_{\mathrm{NSHB}}^2}{(1-\beta)\epsilon^2}$$

▷ $b_{\mathrm{opt}}^\star$ は、$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2\right] \le \epsilon^2$ を達成するために必要な

計算量を最小にするバッチサイズ(数値実験で測定可能)

▷ ϵ は訓練の**停止条件**（例えば、$\epsilon = 0.5$ ）

停止条件

▷ $$C_{\mathrm{SGD}}^2 < \frac{b_{\mathrm{SGD}}^\star \epsilon^2}{\eta},\ C_{\mathrm{SHB}}^2 < \frac{b_{\mathrm{SHB}}^\star \beta(1-\beta)^2\epsilon^2}{\eta(\beta^2-\beta+1)},\ C_{\mathrm{NSHB}}^2 < \frac{b_{\mathrm{NSHB}}^\star (1-\beta)\epsilon^2}{\eta}$$

学習率　　慣性係数

この式で推定可能！

確率的勾配の分散 C_{opt}^2 の推定

▷**CIFAR100 データセットでResNet18を訓練するとき、**

停止条件を $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2\right] \leq \epsilon^2 = (0.5)^2$ **とすると、**

$$b_{\mathrm{SGD}}^{\star} = 2^9,\ b_{\mathrm{SHB}}^{\star} = 2^{10},\ b_{\mathrm{NSHB}}^{\star} = 2^9$$

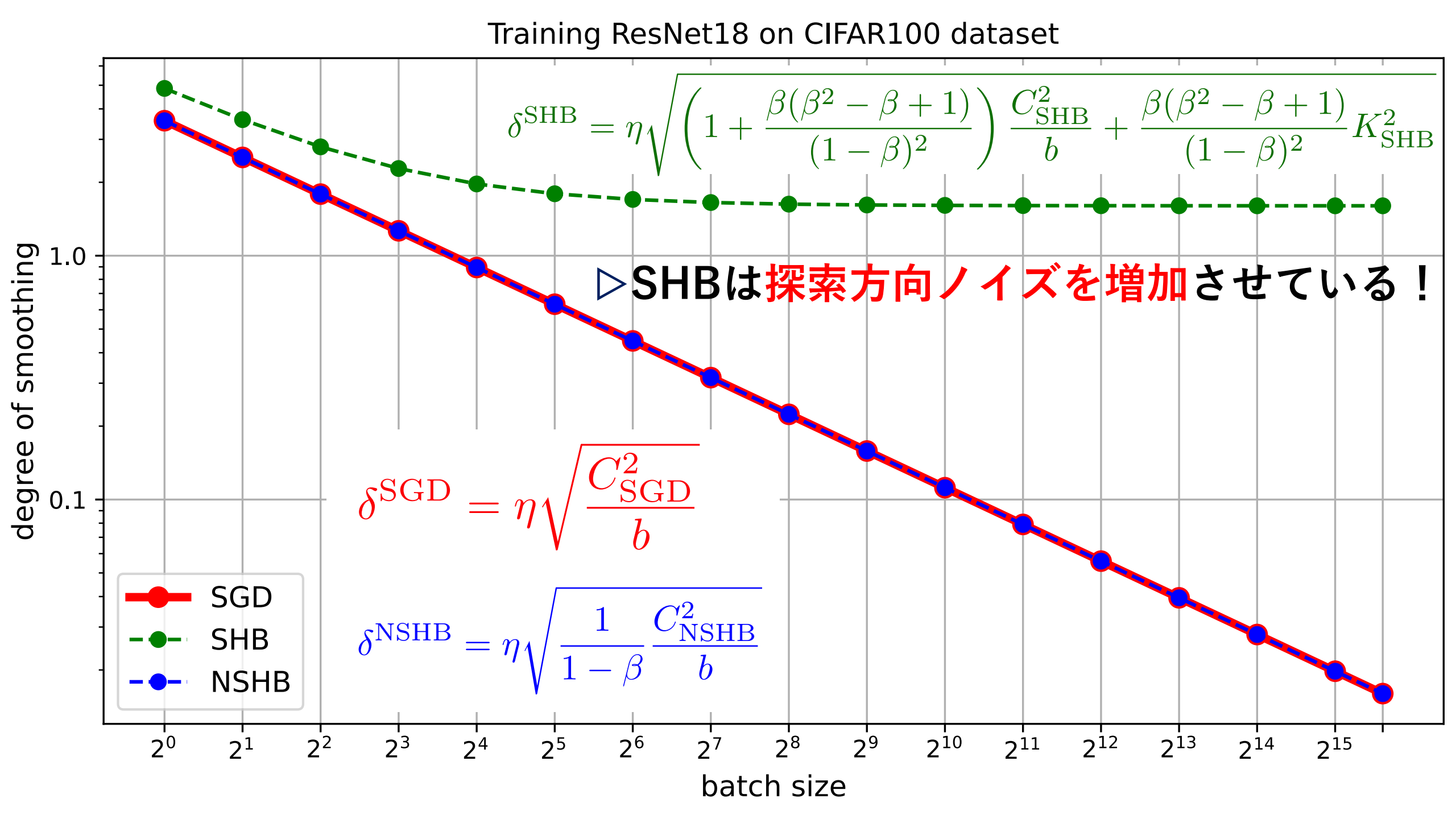
となった。なお、訓練に使用された学習率は $\eta = 0.1$ **で、**

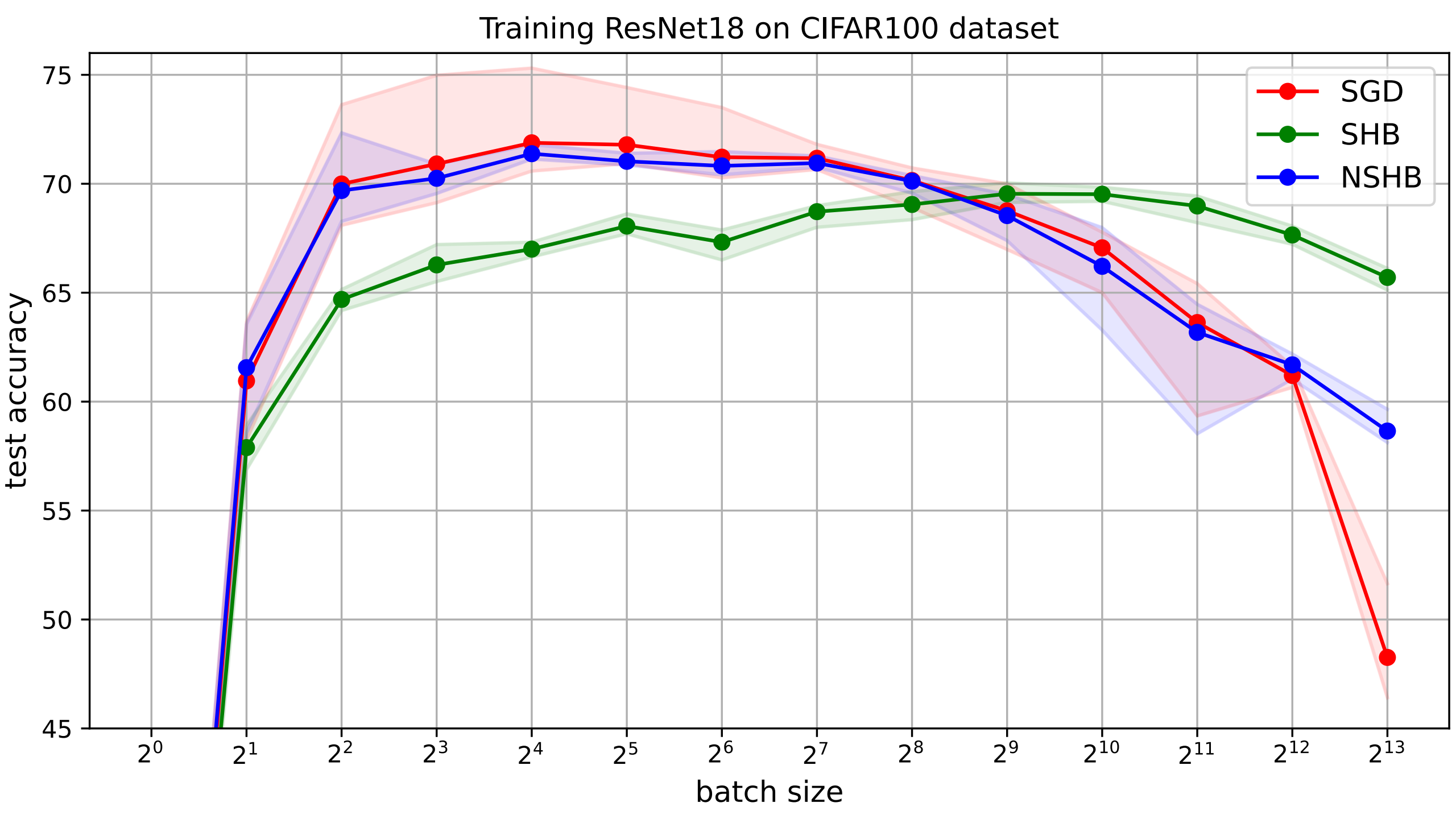
慣性係数は $\beta = 0.9$ **だった。これらの値を代入すると、**

$$C_{\mathrm{SGD}}^2 < \frac{b_{\mathrm{SGD}}^{\star}\epsilon^2}{\eta},\ C_{\mathrm{SHB}}^2 < \frac{b_{\mathrm{SHB}}^{\star}\beta(1-\beta)^2\epsilon^2}{\eta(\beta^2-\beta+1)},\ C_{\mathrm{NSHB}}^2 < \frac{b_{\mathrm{NSHB}}^{\star}(1-\beta)\epsilon^2}{\eta}$$

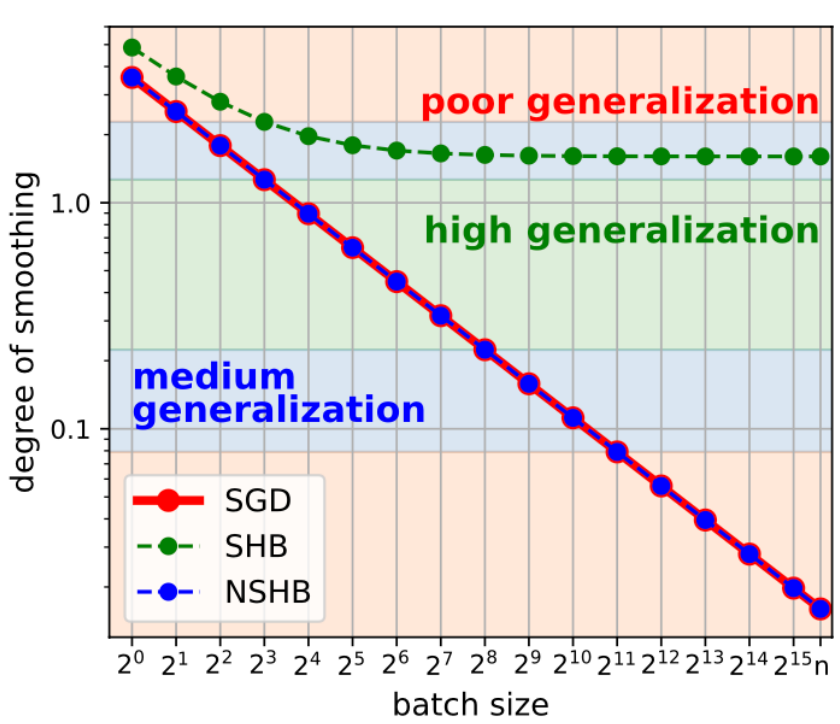
$$= 1280 \qquad = 25.3 \qquad = 128$$

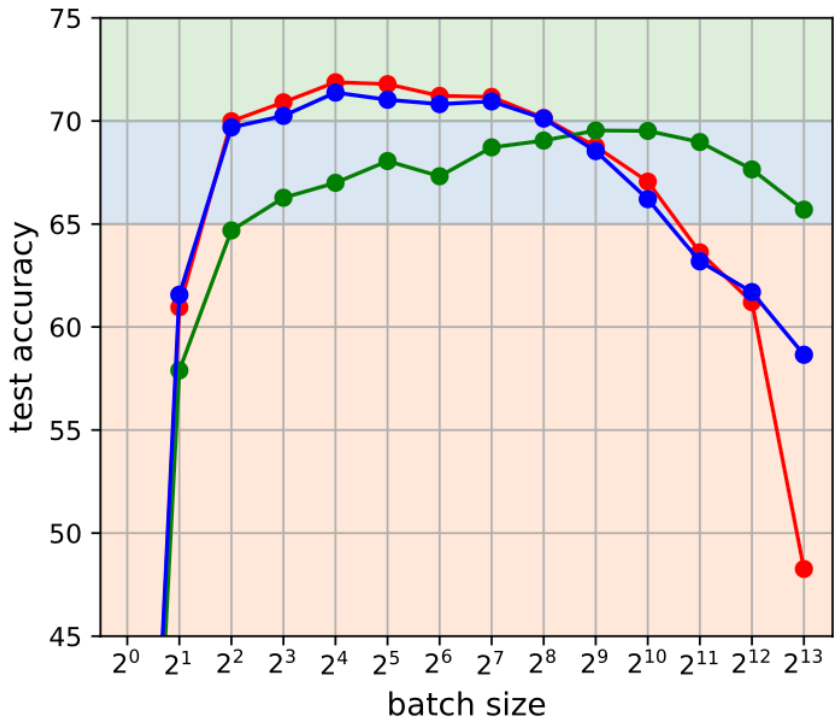
▷**確かにSGD with momentumは勾配ノイズを削減している！**

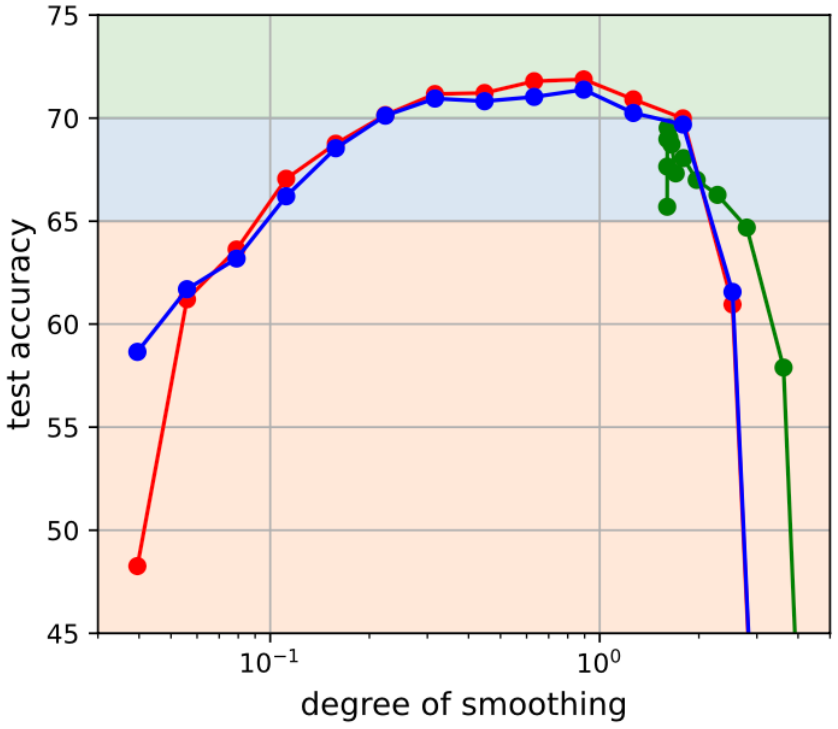

Training ResNet18 on CIFAR100 dataset
$\delta^{\mathrm{SHB}} = \eta\sqrt{\left(1+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}\right)\frac{C_{\mathrm{SHB}}^2}{b}+\frac{\beta(\beta^2-\beta+1)}{(1-\beta)^2}K_{\mathrm{SHB}}^2}$
▷SHBは探索方向ノイズを増加させている！
$\delta^{\mathrm{SGD}} = \eta\sqrt{\frac{C_{\mathrm{SGD}}^2}{b}}$
$\delta^{\mathrm{NSHB}} = \eta\sqrt{\frac{1}{1-\beta}\frac{C_{\mathrm{NSHB}}^2}{b}}$
SGD
SHB
NSHB
degree of smoothing
1.0
0.1
2^0 2^1 2^2 2^3 2^4 2^5 2^6 2^7 2^8 2^9 2^{10} 2^{11} 2^{12} 2^{13} 2^{14} 2^{15}
batch size


Training ResNet18 on CIFAR100 dataset
SGD
SHB
NSHB
test accuracy
batch size
75
70
65
60
55
50
45
2^0
2^1
2^2
2^3
2^4
2^5
2^6
2^7
2^8
2^9
2^{10}
2^{11}
2^{12}
2^{13}

平滑化の度合いと汎化性能の相関







▷バッチサイズが大きくなっても、SHBだけは平滑化の度合いが十分大きいので、汎化性能が悪化しない。

▷NSHBの平滑化の度合いはSGDと同等なので、SGDとNSHBの汎化性能はほぼ一致している。

結論

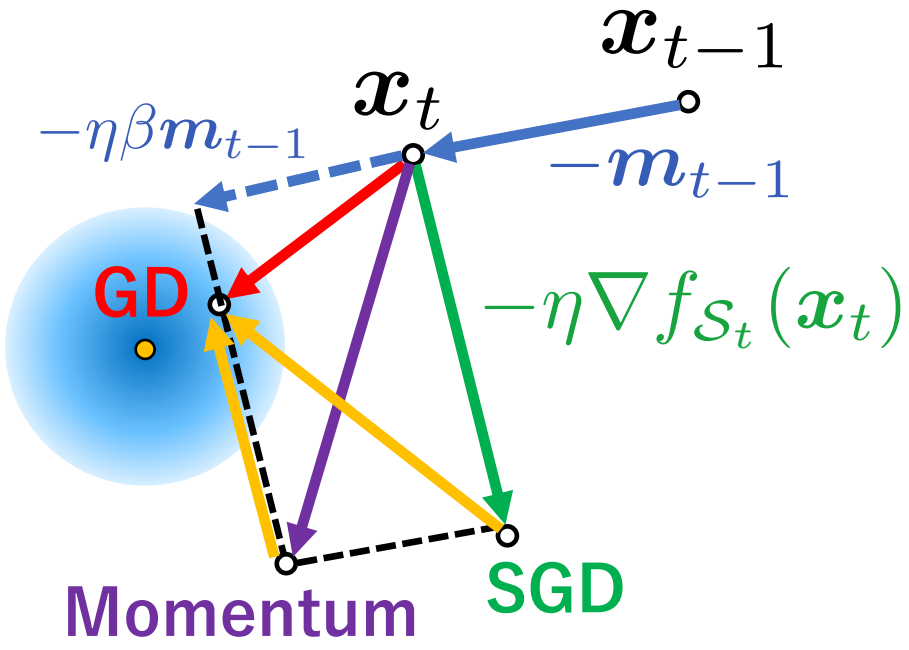
- SGD with momentumには目的関数を平滑化する効果があり、その度合いは学習率、バッチサイズ、慣性係数などで決まる。
- 次の２つの主張が競合しないことを示した。
  - 慣性項は確率的ノイズを削減しているはず。
  - 十分な大きさの確率的ノイズが高い汎化性をもたらすはず。
- 慣性項は勾配ノイズを削減するが、逆に探索方向ノイズを増加させる。そして探索方向ノイズが汎化性能に寄与する。
- 適度な大きさの平滑化の度合いが高い汎化性能をもたらす。
- SHBはバッチサイズが大きいときに有効で、逆にバッチサイズが小さいときは慣性項がある意義がなくなる。
- SHBとNSHBは全くの別物で、SHBの方が優れている。

没・背景：慣性項と確率的ノイズの矛盾

▷慣性項は確率的ノイズを**削減**しているはず。

▷十分な大きさの確率的ノイズが**高い汎化性**をもたらすはず。

▷しかし、Momentumの方がSGDよりも高い汎化性をもたらす。



$$(\mathrm{GD})\ \boldsymbol{y}_t := \boldsymbol{x}_t - \eta\nabla f(\boldsymbol{x}_t)$$

$$(\mathrm{SGD})\ \boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \eta\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)$$

$$(\mathrm{Momentum})\ \boldsymbol{m}_t := \nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) + \beta\boldsymbol{m}_{t-1}$$

$$\hat{\boldsymbol{y}}_t := \boldsymbol{x}_t - \eta\boldsymbol{m}_t$$

SGDの確率的ノイズ

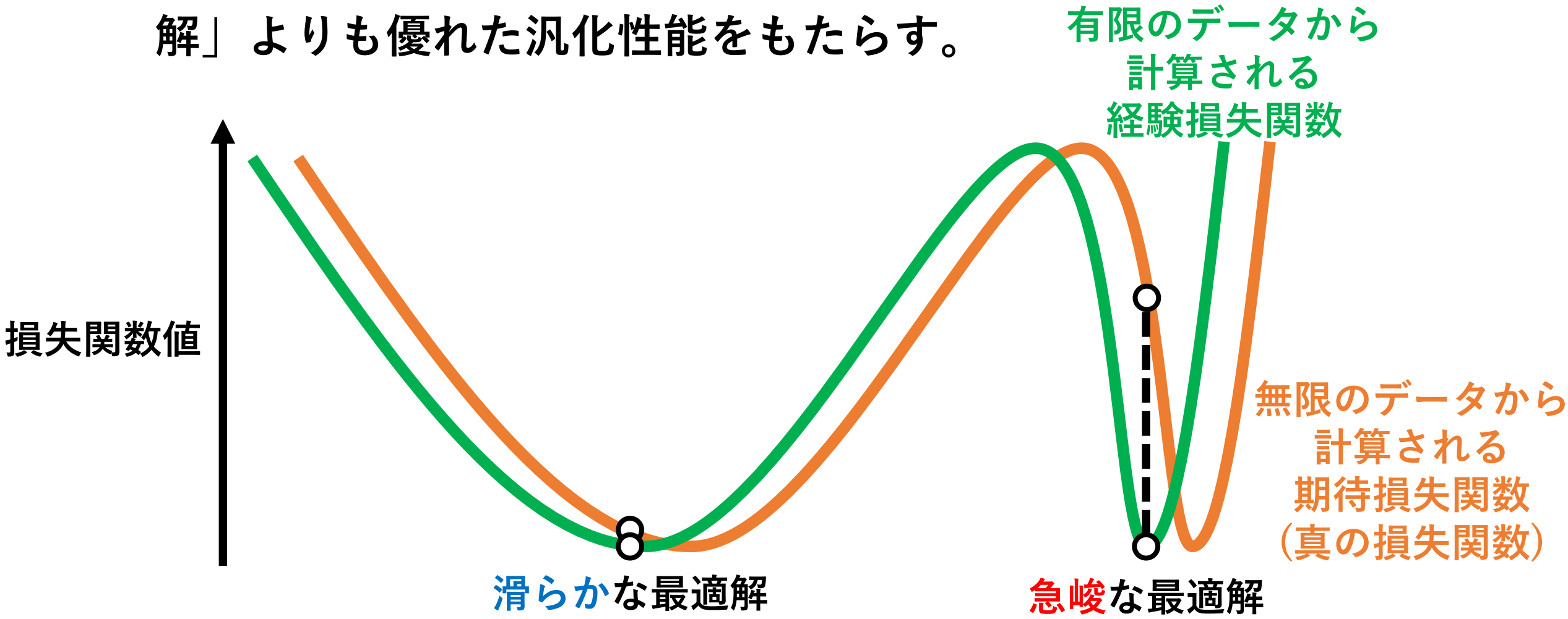
$$\boldsymbol{\omega}_t^{\mathrm{SGD}} := \underbrace{\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)}_{\text{SGDの探索方向}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\text{GDの探索方向}}$$

Momentumの確率的ノイズ

$$\boldsymbol{\omega}_t^{\mathrm{Momentum}} := \underbrace{\boldsymbol{m}_t}_{\substack{\text{Momentum}\\ \text{の探索方向}}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\substack{\text{GD}\\ \text{の探索方向}}}$$

付録：汎化性能と経験損失関数の形状

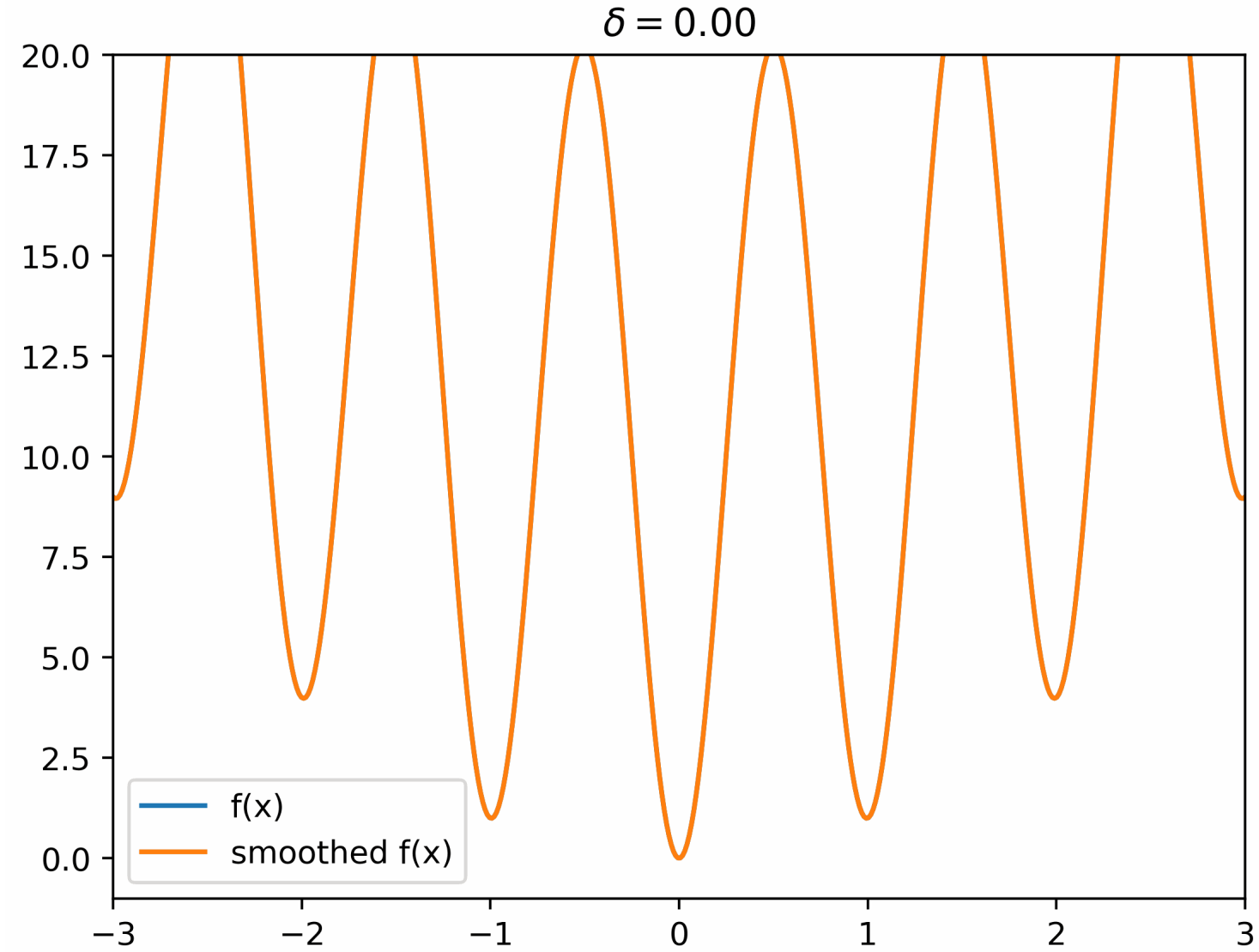
▷「その近傍が滑らかな最適解」は、「その近傍が急峻な最適解」よりも優れた汎化性能をもたらす。



付録：関数の平滑化

▷**目的関数の平滑化**

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim\mathcal{N}\left(\mathbf{0},\frac{1}{\sqrt{d}}I_d\right)}[f(\boldsymbol{x}-\delta\boldsymbol{u})]$$



▷1変数のRastrigin's 関数

$$f(x) = x^2 - 10\cos(2\pi x) + 10$$

$$|\hat{f}_\delta(\boldsymbol{x}) - f(\boldsymbol{x})| \leq |\delta| L_f$$