確率的勾配降下法の平滑化効果を利用した段階的最適化手法によるディープニューラルネットワークの大域的最適化

日本オペレーションズリサーチ2024年春季研究発表会
3/7(火) 10:00～10:20

05001736 明治大学 佐藤尚樹
01016200 明治大学 飯塚秀明

背景：機械学習（画像分類）

入力

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

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

$d = 512 \times 512 \times 3$

$N = 3000000$

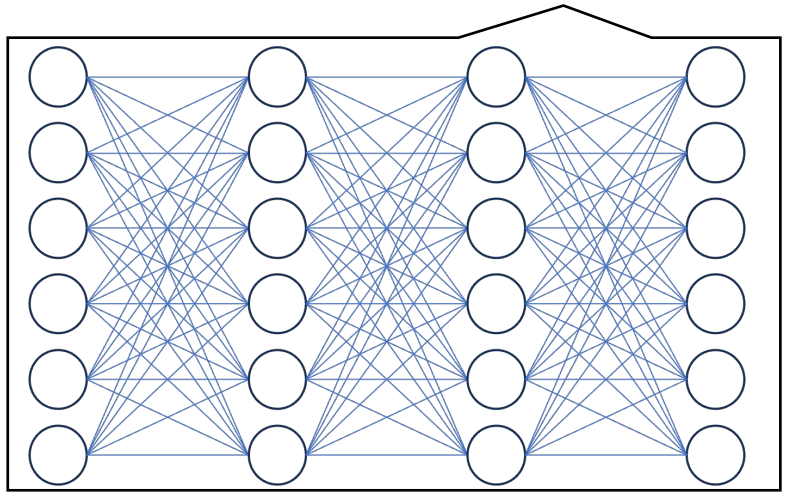
データセット

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

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

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

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

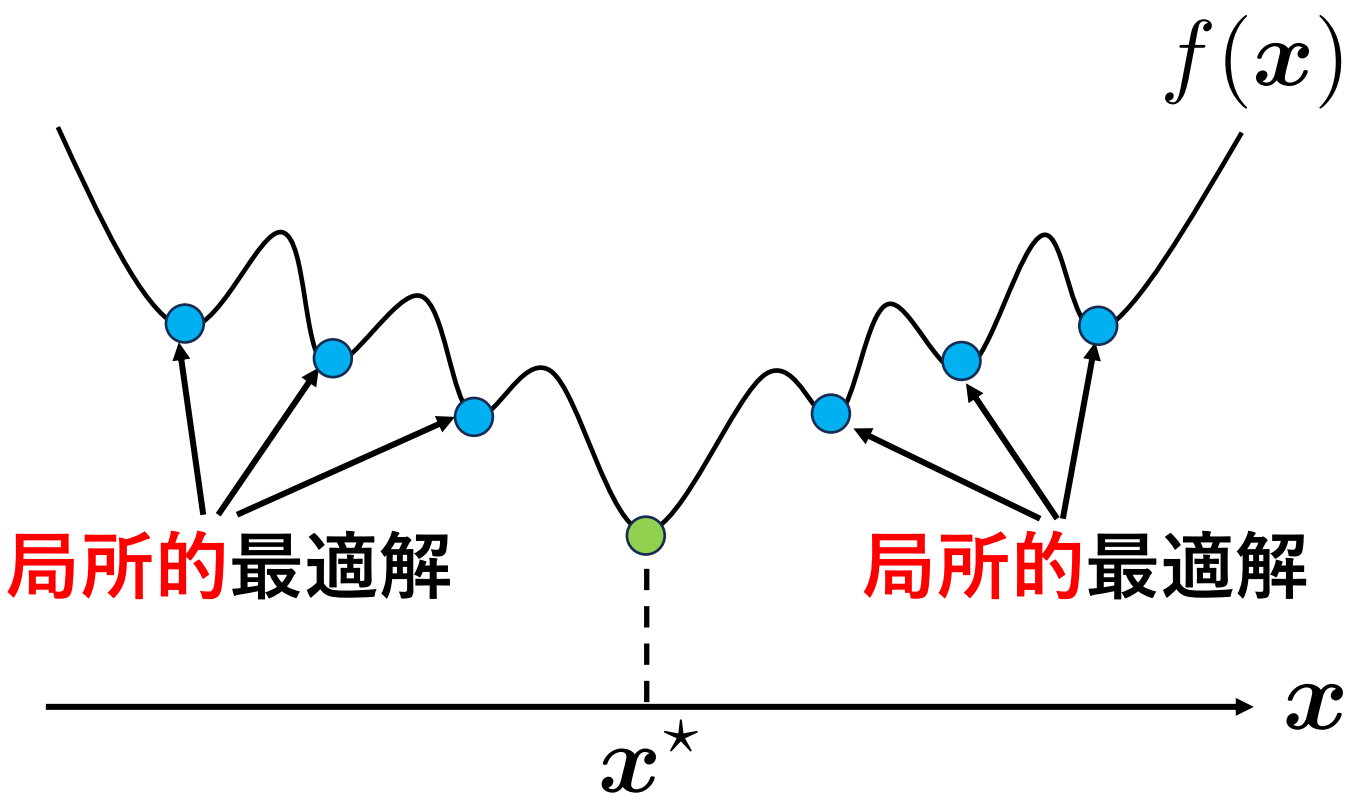
ミニバッチ確率的勾配

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：大域的最適化の難しさ

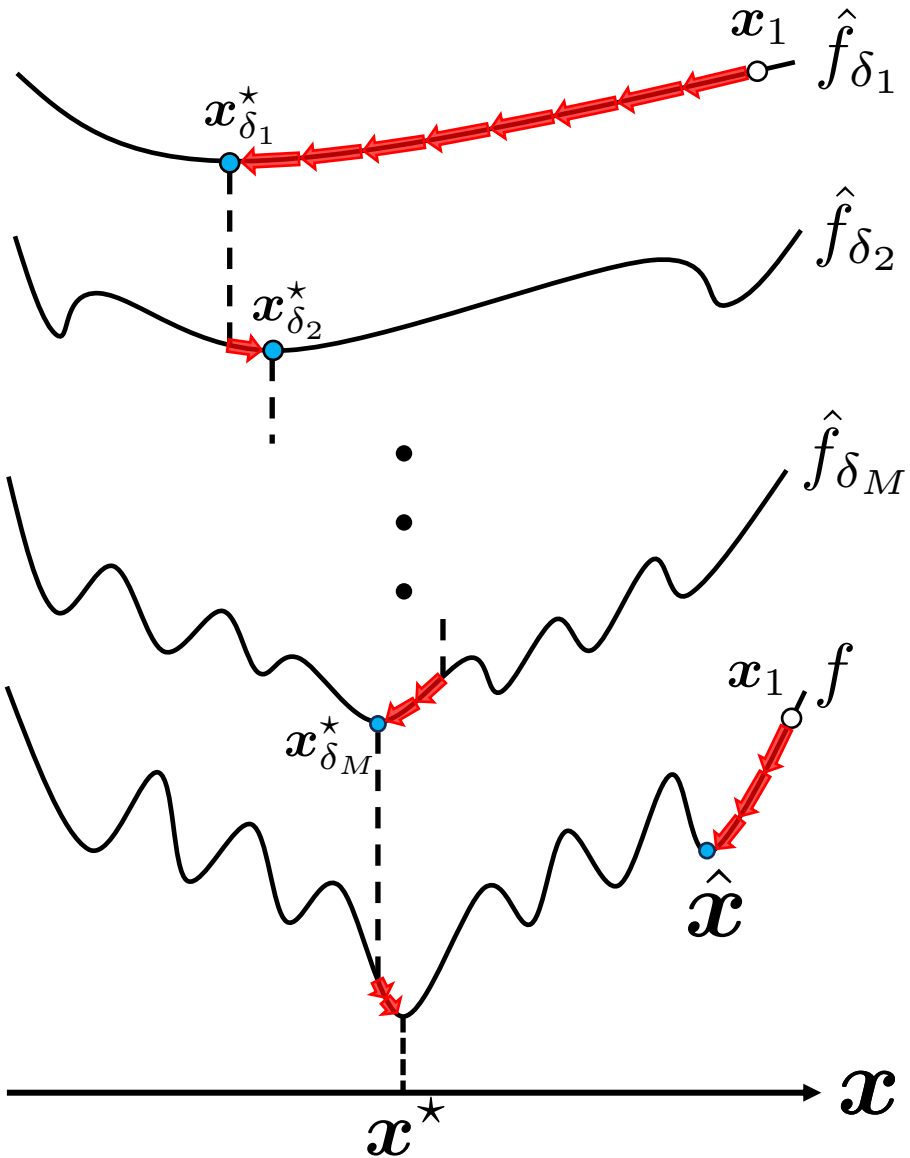


機械学習に使用される最適化アルゴリズムは、全て勾配を利用する手法

勾配に頼っている手法は、山を登れない

非凸関数の大域的最適解を見つけることは難しい

背景：段階的最適化 (Graduated Optimization)



▷1987年に提案された大域的最適化手法

▷徐々に小さくなるノイズで平滑化された関数の列を順番に最適化する

▷目的関数の平滑化

大きさδのノイズで平滑化された関数は、

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim B(\mathbf{0};1)}\left[f(\boldsymbol{x}-\delta\boldsymbol{u})\right]$$

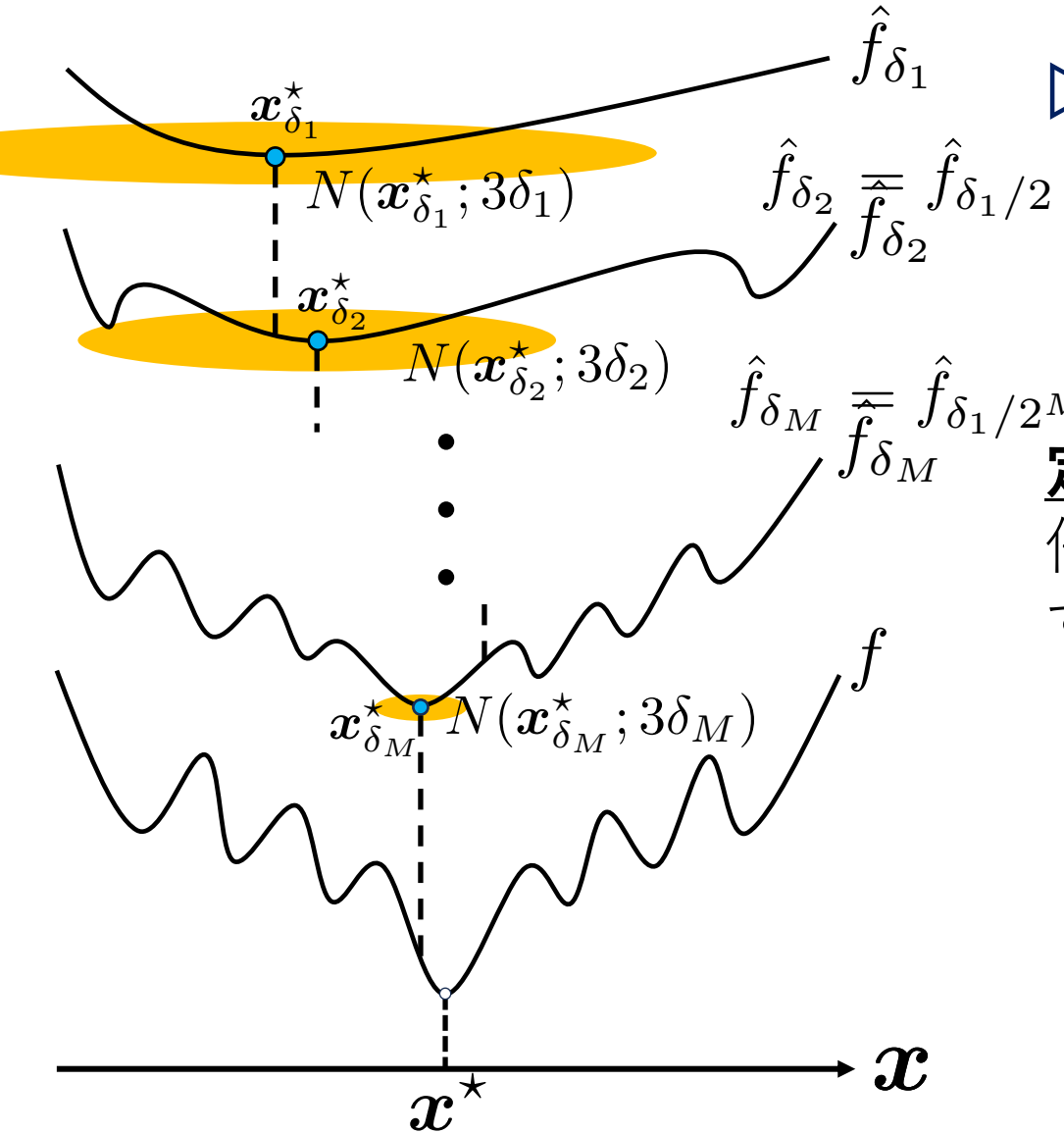
↳中心0, 半径1の閉球

$$= \int p(\boldsymbol{u}) f(\boldsymbol{x}-\delta\boldsymbol{u}) d\boldsymbol{u}$$

計算可能　　計算不可能

背景：σ-nice 関数[2]

[2] E. Hazan et al. “On Graduated Optimization for Stochastic Non-Convex Problems”
In *Proceedings of the 33th International Conference on Machine Learning*, pp1833-1841, 2016



▷**段階的最適化アルゴリズムが大域的最適解に収束するために必要な性質を満たす特別な非凸関数の族。**

定義

任意の非凸関数 $f: \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f は σ-nice 関数であるという。

（i）任意の $\delta > 0$ と $\boldsymbol{x}^\star_\delta$ に対して、$\boldsymbol{x}^\star_{\delta/2}$ が存在して、

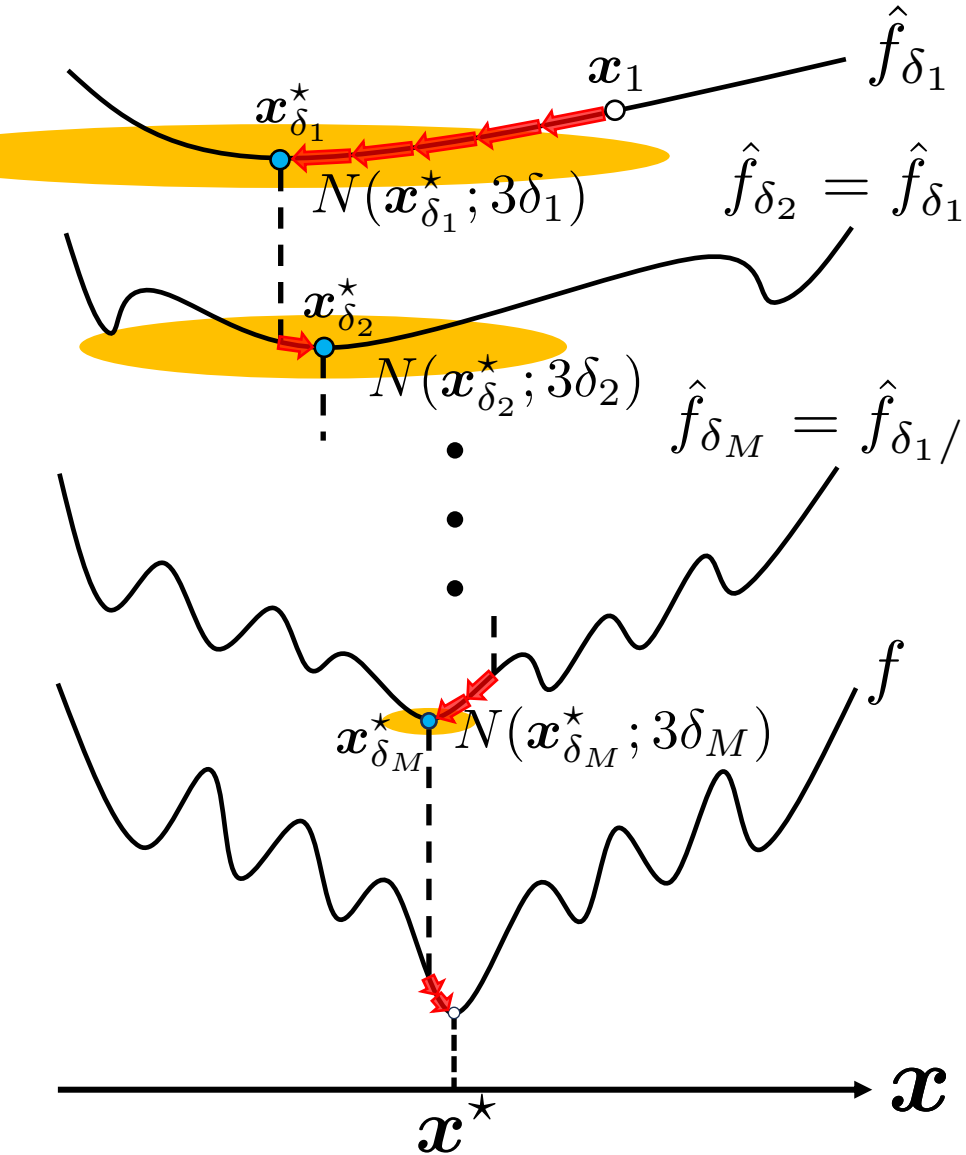
$$\left\| \boldsymbol{x}^\star_\delta - \boldsymbol{x}^\star_{\delta/2} \right\| \leq \frac{\delta}{2}$$

を満たす。

（ii）任意の $\delta > 0$ に対して、関数 $\hat{f}_\delta(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}^\star_\delta; 3\delta)$ で σ–強凸となる。

背景：σ-nice 関数[2]

[2] E. Hazan et al. “On Graduated Optimization for Stochastic Non-Convex Problems”
In *Proceedings of the 33th International Conference on Machine Learning*, pp1833-1841, 2016



Algorithm 1 Graduated optimization **（実現不可能）**

Require: $\delta_1 > 0, x_1 \in N(x^\star_{\delta_1}; 3\delta_1)$
  for $m = 1$ to M **do**
    $T_m := \sigma\delta_m^2/32$ ←反復回数を決定
    $x_{m+1} := \mathrm{SGD}(T_m, x_m, \hat{f}_{\delta_m})$ ←←←平滑化された関数をSGD等で最適化
    $\delta_{m+1} := \delta_m/2$ ←ノイズレベルを更新
  end for
  return x_{M+1}

定理5.1 [2]

σ-nice 関数 $f : \mathbb{R}^d \to \mathbb{R}$ に対して、Algorithm 1を適用すると、$\mathcal{O}\left(1/\epsilon^2\right)$ 回の反復で f の大域的最適解 $x^\star$ の ϵ-近傍に到達する。

背景：SGDの確率的ノイズと平滑化

▷**確率的勾配降下法は、なぜか汎化性の高い解に収束する。**
特に画像分類では、Adamよりも優れることがある。

付録：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 $g\colon \mathbb{R}^d \to \mathbb{R}$

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

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

Deep Neural Network

訓練済み

＝パラメータ調整済

猫

訓練損失: 0.001

訓練精度: 99.9%

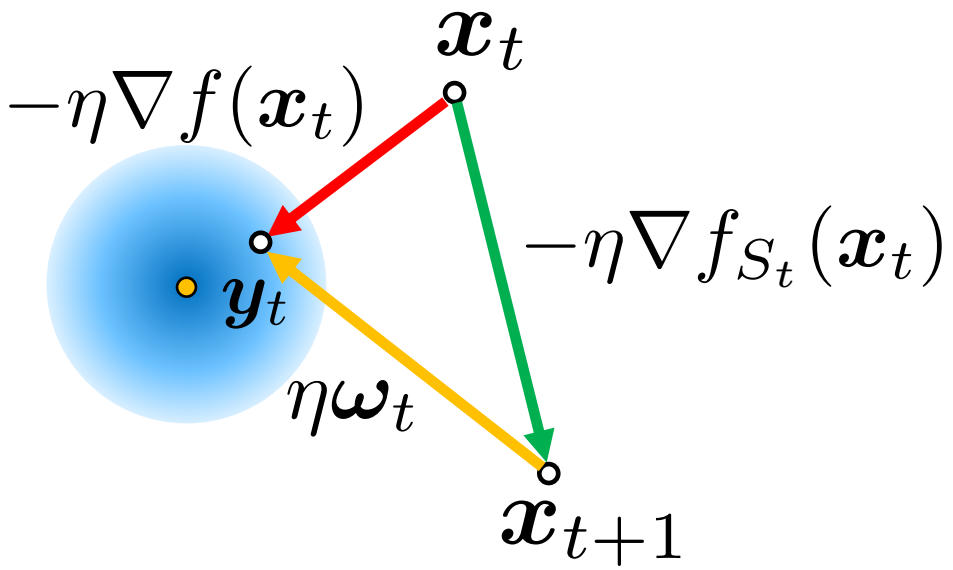
これも猫！

テスト精度: 75%

背景：SGDの確率的ノイズと平滑化

▷**確率的勾配降下法(SGD)は、なぜか汎化性の高い解に収束する。**
特に画像分類では、Adamよりも優れることがある。

▷**SGDの確率的ノイズが役立っている？[1]**



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

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

$$(\text{確率的ノイズ})\ \boldsymbol{\omega}_t := \underbrace{\nabla f_{S_t}(\boldsymbol{x}_t)}_{\text{SGDの探索方向}} - \underbrace{\nabla f(\boldsymbol{x}_t)}_{\substack{\text{GDの探索方向} \\ \text{（最急降下方向）}}}$$

[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**

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u} \sim B(\mathbf{0};1)} \left[f(\boldsymbol{x} - \delta \boldsymbol{u}) \right]$$

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

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

▷**”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

動機

▷ σ-nice 関数がどれくらい特別な関数なのか分かっていない。
そもそも存在するかどうかも不明。

➡ 任意の非凸関数が σ-nice 関数になるための十分条件を明らかにしたい。

▷SGDが持つ確率的ノイズには関数を平滑化する効果がある。

➡ 平滑化の度合い, すなわちノイズレベル δ が何によって定まるのかを明らかにしたい。

➡ 訓練中にノイズレベル δ が徐々に小さくなるようにすることで、SGDを利用した段階的最適化アルゴリズムを構築したい。

➡ DNNを含む非凸関数の大域的最適化を実現したい。

貢献

- σ-nice 関数を拡張したnew σ-nice 関数を定義し、任意の関数がnew σ-nice 関数となるための**十分条件**を示した。
- SGDの確率的ノイズによる平滑化の度合いは、

$$\delta = \eta \sqrt{\frac{C^2}{b}}$$

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**、C^2 は**確率的勾配の分散**とする。

- SGDの平滑化特性を利用した**暗黙的な**段階的最適化アルゴリズムを提案し、それが $\mathcal{O}\left(1/\epsilon^{\frac{1}{p}}\right)$ $(p \in (0, 1])$ 回の反復でnew σ-nice関数の**大域的最適解** $x^\star$ の ϵ-近傍に到達できることを示した。

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_g**-平滑とする。**

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

(A2)関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ **は**L_f**-リプシッツ関数とする。**

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

(A3) $(\boldsymbol{x}_t)_{t\in\mathbb{N}} \subset \mathbb{R}^d$ **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^2 **が存在するとする。**

$$\mathbb{E}_{\xi_t}\left[\left\|\mathsf{G}_{\xi_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)\right\|^2\right] \leq C^2$$ ←確率的勾配の分散

準備：仮定

仮定続き

(A4) 時刻$t \in \mathbb{N}$で、全勾配∇fは<u>ミニバッチ</u>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

補題2.1

仮定(A3)(ii)と(A4)が成り立つとすると、任意の$t \in \mathbb{N}$に対して、

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

SGDの探索方向

GDの探索方向（最急降下方向）

が成り立つ。

New σ-nice関数

定義 (σ-nice関数)
任意の非凸関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f は σ-nice関数であるという。

（i）任意の $\delta > 0$ と $\boldsymbol{x}_\delta^\star$ に対して、

$$\left\|\boldsymbol{x}_\delta^\star - \boldsymbol{x}_{\delta/2}^\star\right\| \leq \frac{\delta}{2} \quad \left(\gamma_m := \frac{1}{2}\right)$$

が成り立つ。

（ii）任意の $\delta > 0$ に対して、関数 $\hat{f}_\delta(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}_\delta^\star; 3\delta)$ でσ–強凸となる。

定義 (New σ-nice 関数)
任意の非凸リプシッツ関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f はnew σ-nice 関数であるという。

（i）任意の $\delta_m \in \mathbb{R}$ と $\boldsymbol{x}_{\delta_m}^\star$ に対して、

$$\left\|\boldsymbol{x}_{\delta_m}^\star - \boldsymbol{x}_{\delta_{m+1}}^\star\right\| \leq |\delta_m| - |\delta_{m+1}|$$

が成り立つ。ただし、

$$\delta_{m+1} := \gamma_m \delta_m \ (\gamma_m \in (0,1))$$

とする。

（ii）任意の $\delta_m \in \mathbb{R}$ に対して、関数 $\hat{f}_{\delta_m}(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}^\star; d_m\delta_m)$ でσ–強凸となる。

New σ-nice関数

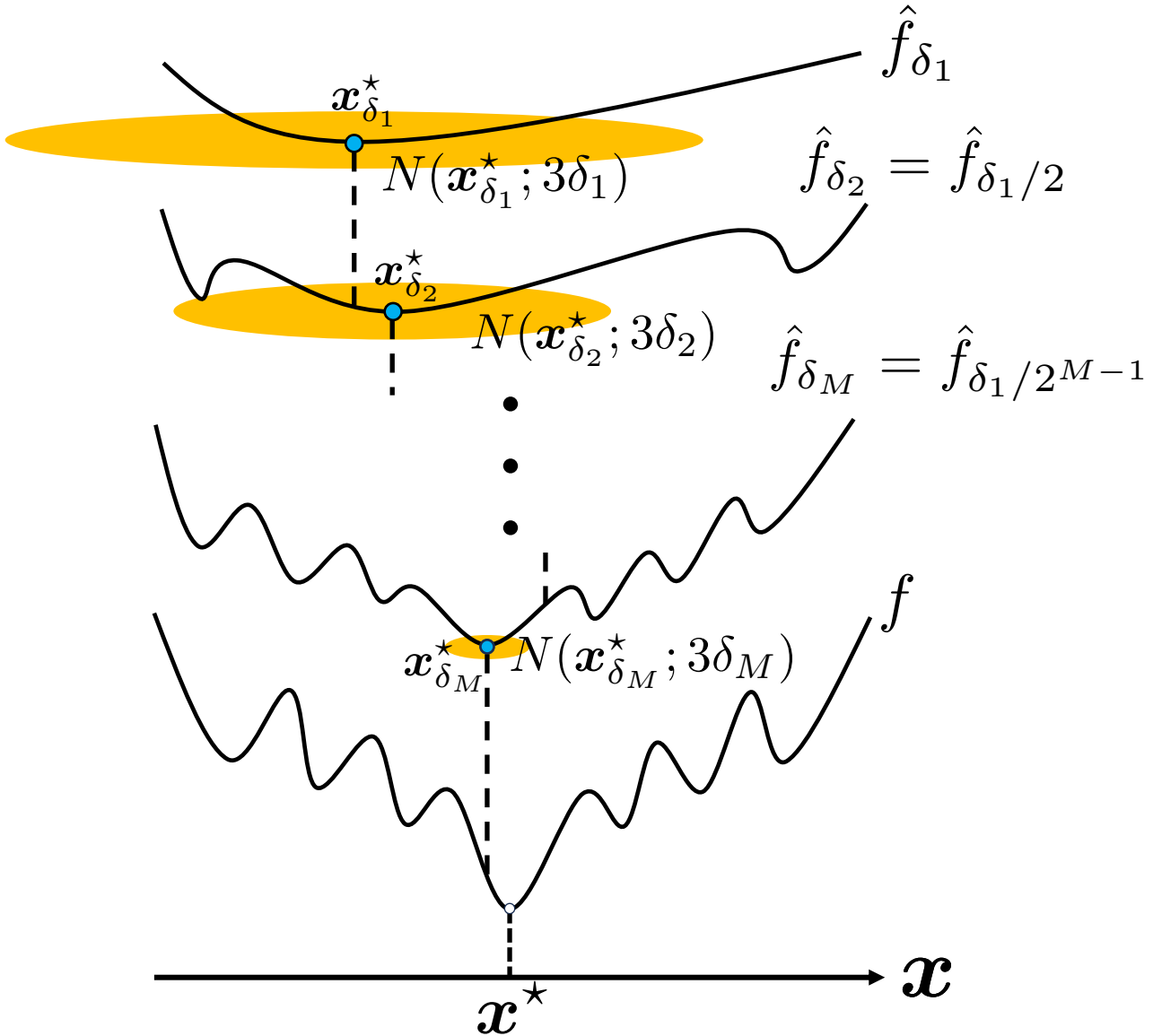
定義 (σ-nice関数)
任意の非凸関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f は σ-nice関数であるという。

（i）任意の $\delta > 0$ と $\boldsymbol{x}^\star_\delta$ に対して、

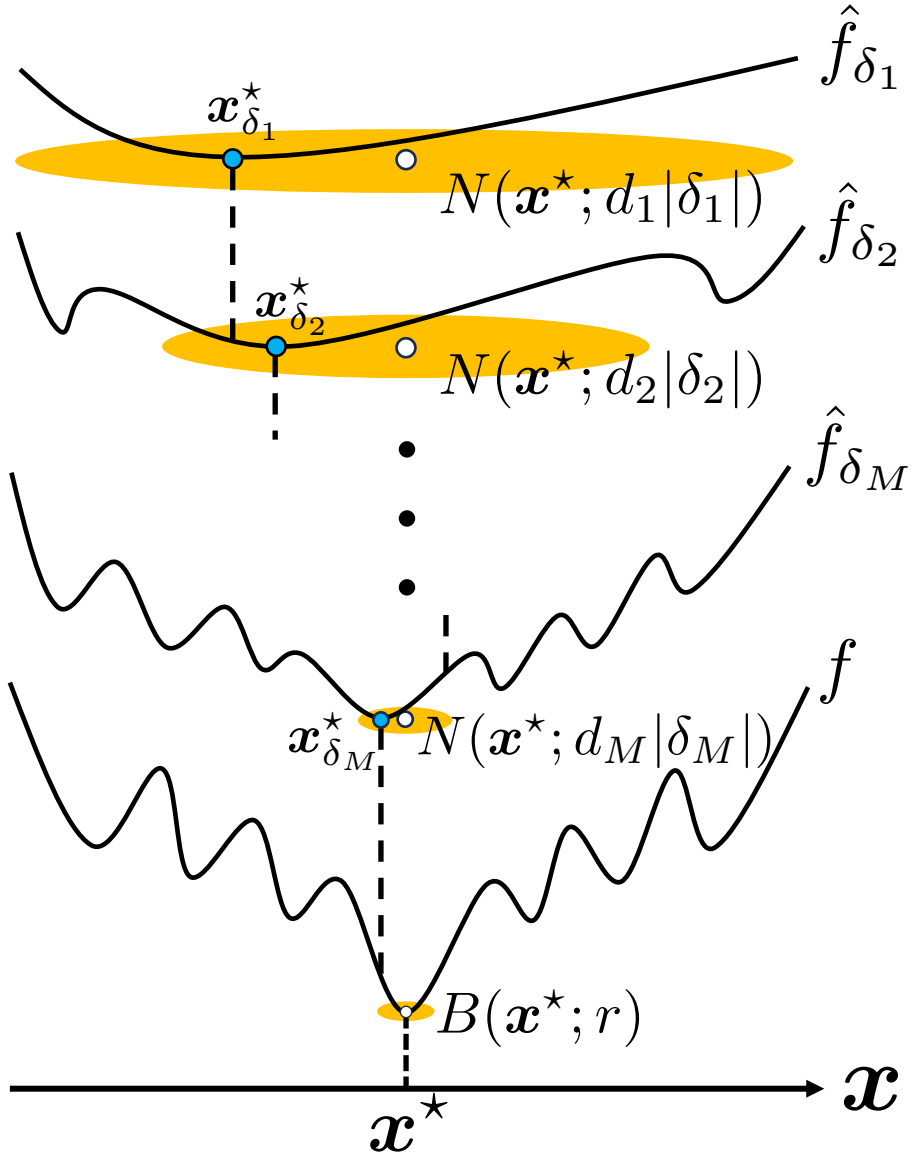
$$\left\|\boldsymbol{x}^\star_\delta - \boldsymbol{x}^\star_{\delta/2}\right\| \leq \frac{\delta}{2} \quad \left(\gamma_m := \frac{1}{2}\right)$$

が成り立つ。

（ii）任意の $\delta > 0$ に対して、関数 $\hat{f}_\delta(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}^\star_\delta; 3\delta)$ で σ–強凸となる。



New σ-nice関数



定義 (New σ-nice 関数)

任意の非凸リプシッツ関数 $f: \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f はnew σ-nice 関数であるという。

（i）任意の $\delta_m \in \mathbb{R}$ と $x^\star_{\delta_m}$ に対して、

$$\left\| x^\star_{\delta_m} - x^\star_{\delta_{m+1}} \right\| \leq |\delta_m| - |\delta_{m+1}|$$

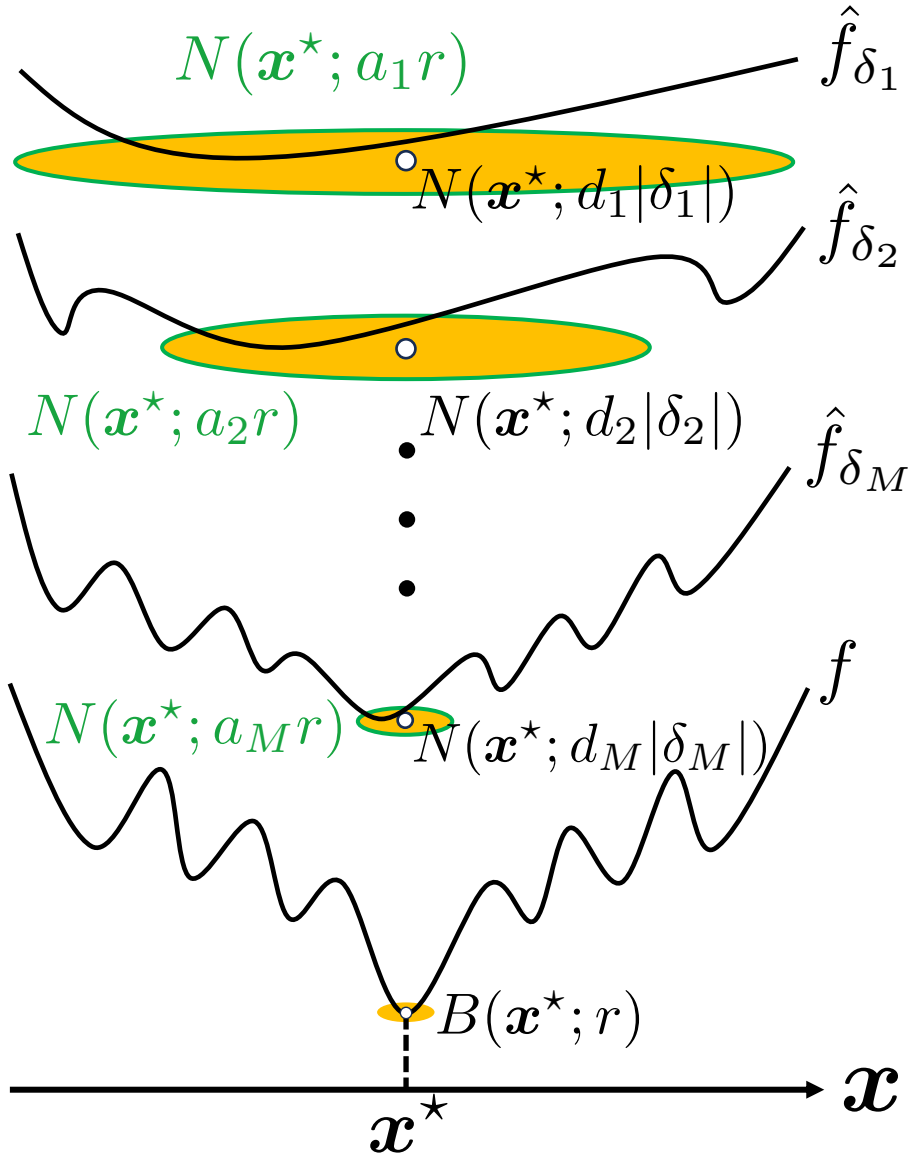
が成り立つ。ただし、

$$\delta_{m+1} := \gamma_m \delta_m \ (\gamma_m \in (0,1))$$

とする。

（ii）任意の $\delta_m \in \mathbb{R}$ に対して、関数 $\hat{f}_{\delta_m}(x)$ は近傍 $N(x^\star; d_m|\delta_m|)$ で σ–強凸となる。

New σ-nice性の十分条件



命題3.2

関数$f\colon \mathbb{R}^d \to \mathbb{R}$が十分小さい$r > 0$に対して閉球$B(\boldsymbol{x}^\star; r)$で$\sigma$–強凸だとすると、関数$f$がnew σ-nice関数となるための十分条件は、ノイズレベルδ_mが次の条件を満たすことである。

$$\frac{2\max\left\{\left\|\nabla \hat{f}_{\delta_m}(\boldsymbol{x}^\star)\right\|, \left\|\nabla \hat{f}_{\delta_{m+1}}(\boldsymbol{x}^\star)\right\|\right\}}{\sigma(1-\gamma_m)} \leq |\delta_m| = |\delta^-|$$

δ^-が存在する確率$p(a_m)$

$$0 < p(a_m) := \frac{2\arccos\left(\frac{r\sqrt{a_m^2-1}}{\|\boldsymbol{x}^\star - \boldsymbol{x}\|\|\boldsymbol{u}_m\|}\right)}{\pi} < 1$$

➡ a_mが大きいとき、$p(a_m)$はほぼ0

背景：SGDの確率的ノイズと平滑化(再掲)

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

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

▷**目的関数の平滑化**

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim B(\mathbf{0};1)}\left[f(\boldsymbol{x}-\delta\boldsymbol{u})\right]$$

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

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

▷**”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

SGDの平滑化特性

補題2.1

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

▷目的関数の平滑化

$$\hat{f}_\delta(\boldsymbol{x}):=\mathbb{E}_{\boldsymbol{u}\sim B(\mathbf{0};1)}\left[f(\boldsymbol{x}-\delta\boldsymbol{u})\right]$$

▷ $\boldsymbol{\omega}_t:=\nabla f_{S_t}(\boldsymbol{x}_t)-\nabla f(\boldsymbol{x}_t)$ と補題2.1から、

$$\boldsymbol{\omega}_t=\frac{C}{\sqrt{b}}\boldsymbol{u}_t\ (\boldsymbol{u}_t\sim B(\mathbf{0};1))$$

↑ 中心0, 半径1の閉球

が成り立つ。したがって、

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_{t+1}\right]&=\boldsymbol{y}_t-\eta\nabla\mathbb{E}_{\boldsymbol{\omega}_t}\left[f(\boldsymbol{y}_t-\eta\boldsymbol{\omega}_t)\right]\\&=\boldsymbol{y}_t-\eta\nabla\mathbb{E}_{\boldsymbol{u}_t\sim B(\mathbf{0};1)}\left[f\left(\boldsymbol{y}_t-\frac{\eta C}{\sqrt{b}}\boldsymbol{u}_t\right)\right]\\&=\boldsymbol{y}_t-\eta\nabla\hat{f}_{\frac{\eta C}{\sqrt{b}}}(\boldsymbol{y}_t)\end{aligned}$$

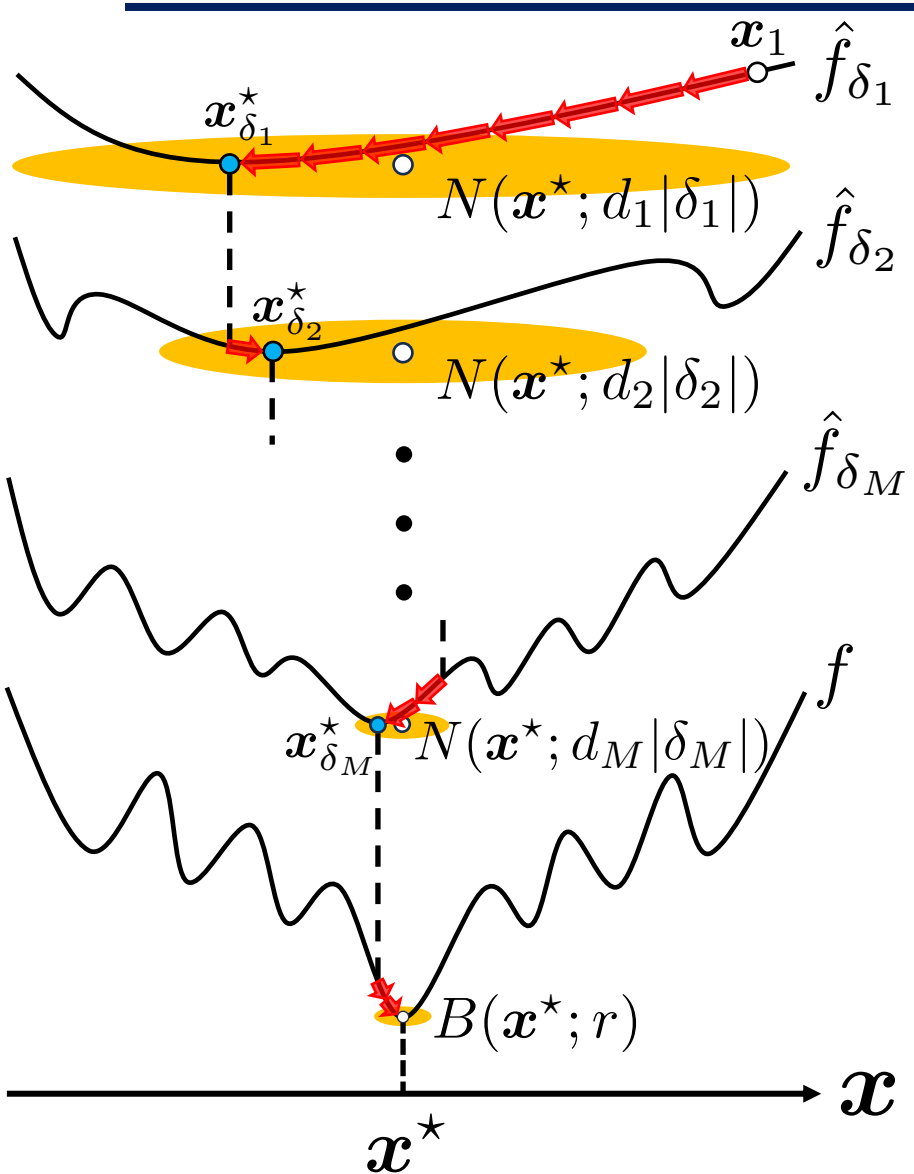
大きさ $\eta C/\sqrt{b}$ のノイズで平滑化された関数

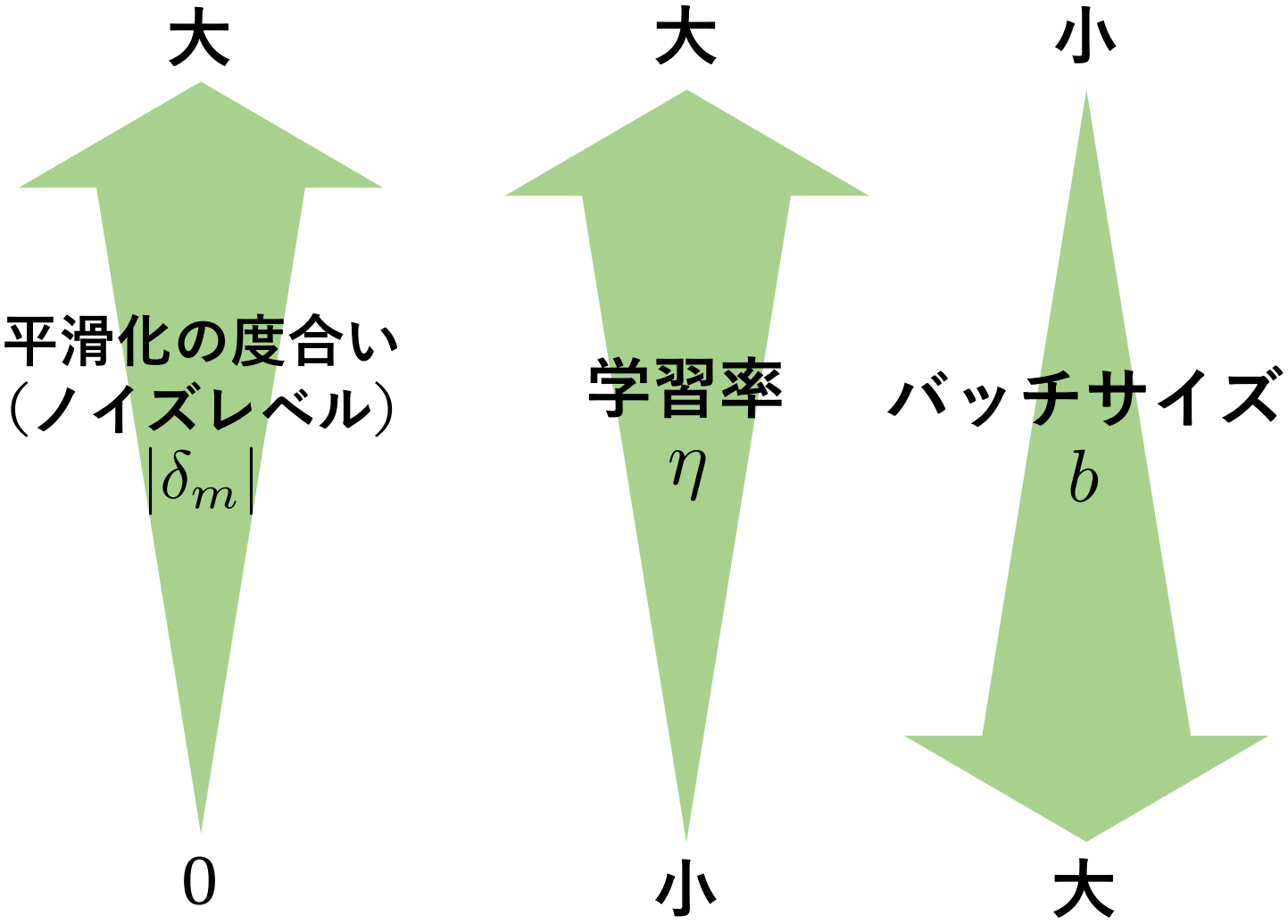
が成り立つ。

$$\delta=\eta\sqrt{\frac{C^2}{b}}$$

SGDの平滑化特性

$$\delta = \eta\sqrt{\frac{C^2}{b}}$$





暗黙的な段階的最適化アルゴリズム

Algorithm 3 Implicit Graduated Optimization

Require: $\epsilon > 0, p \in (0,1], \bar{d} > 0, \boldsymbol{x}_1, \eta_1, b_1$

$\delta_1 := \frac{\eta_1 C}{\sqrt{b_1}}$

$\alpha_0 := \min\left\{\frac{\sqrt{b_1}}{4L_f\eta_1 C(1+\bar{d})}, \frac{\sqrt{b_1}}{\sqrt{2\sigma}\eta_1 C}\right\}, M^p := \frac{1}{\alpha_0\epsilon}$

for $m = 1$ to $M+1$ **do**

  if $m \neq M+1$ **then**

    $\epsilon_m := \sigma^2\delta_m^2,\ T_F := H_4/(\epsilon_m - H_3\eta_m)$

    $\gamma_m := \frac{(M-m)^p}{\{M-(m-1)\}^p}$

    $\kappa_m/\sqrt{\lambda_m} = \gamma_m\ (\kappa_m \in (0,1], \lambda_m \geq 1)$

  end if

  $\boldsymbol{x}_{m+1} := \mathrm{GD}(T_F, \boldsymbol{x}_m, \hat{f}_{\delta_m}, \eta_m, b_m)$

  $\eta_{m+1} := \kappa_m\eta_m, b_{m+1} := \lambda_m b_m$

  $\delta_{m+1} := \frac{\eta_{m+1}C}{\sqrt{b_{m+1}}}$

end for

return $\boldsymbol{x}_{M+2}$

定理3.4

関数$f: \mathbb{R}^d \to \mathbb{R}$が$L_f$-リプシッツ new σ-nice 関数だとすると、アルゴリズム3は、$\mathcal{O}\left(1/\epsilon^{\frac{1}{p}}\right)(p \in (0,1])$回の反復で、関数$f$の**大域的最適解**$\boldsymbol{x}^\star$の$\epsilon$-近傍に到達する。

Algorithm 4 Gradient Descent (GD)

Require: $T_F, \hat{\boldsymbol{x}}_1, F, \eta, b$

  for $t = 1$ to T_F **do**

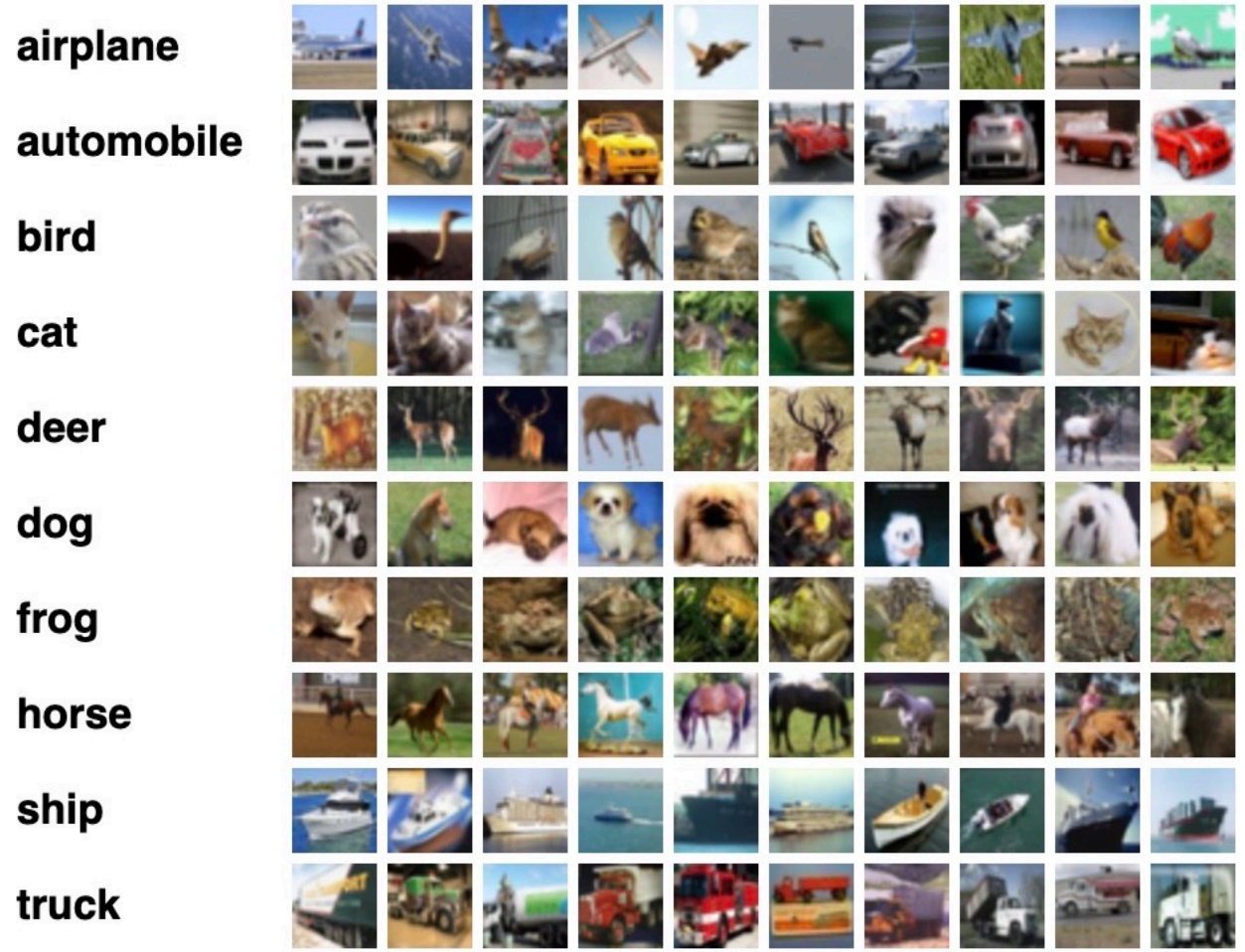
    $\hat{\boldsymbol{x}}_{t+1} := \hat{\boldsymbol{x}}_t - \eta\nabla F(\boldsymbol{x}_t)$

  end for

  return $\hat{\boldsymbol{x}}_{T_F+1} = \mathrm{GD}(T_F, \hat{\boldsymbol{x}}_1, F, \eta)$

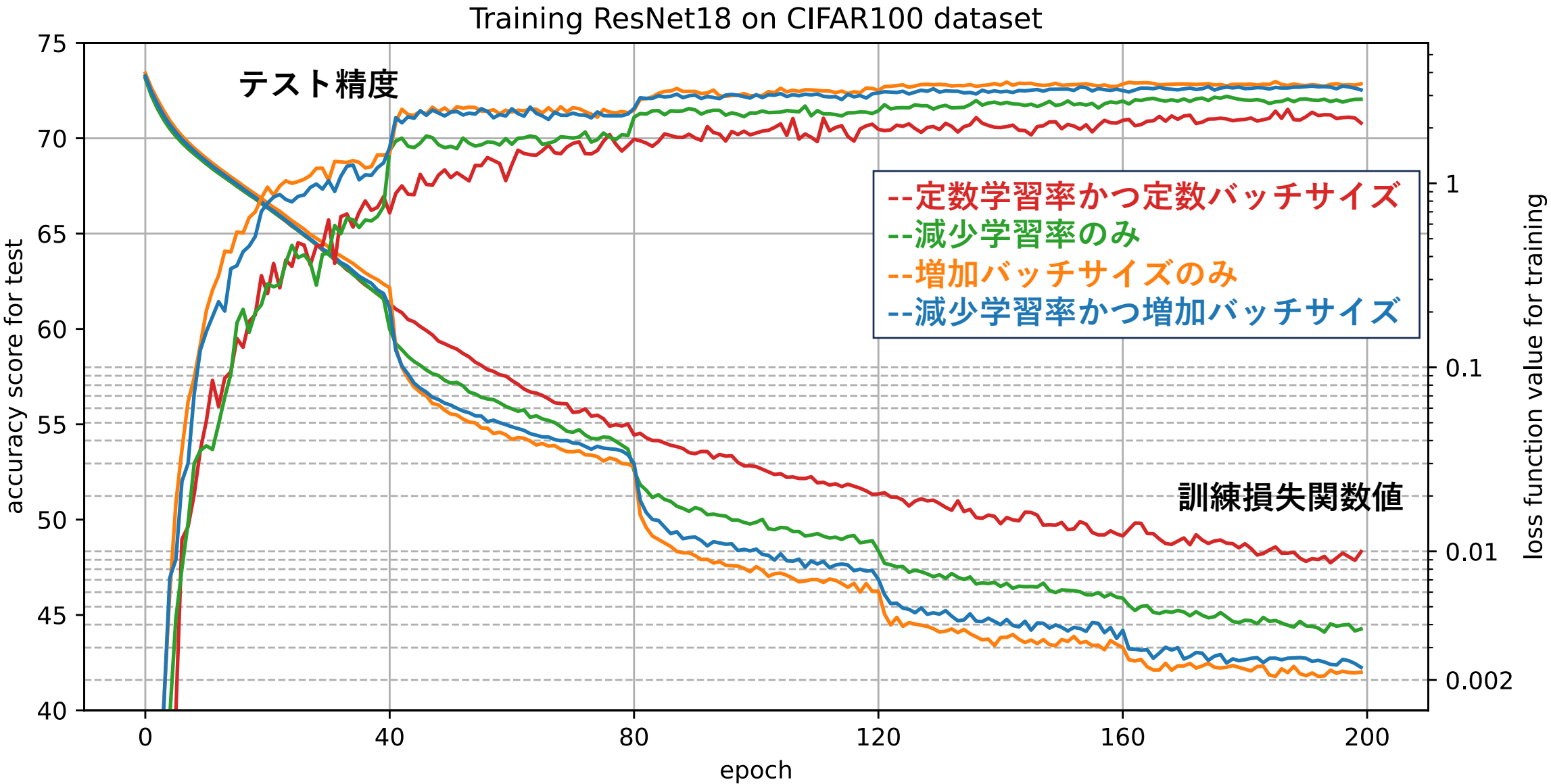
数値実験

- ▷モデル: ResNet18
- ▷データセット: CIFAR100
  - ▷100クラスの画像分類
  - ▷5万枚の訓練データ
  - ▷1万枚のテストデータ
  - ▷画像サイズは32×32×3
- ▷交差エントロピー誤差
- ▷NVIDIA GeForce RTX4090



数値実験ー暗黙的な段階的最適化

$$\delta = \eta \sqrt{\frac{C^2}{b}}$$

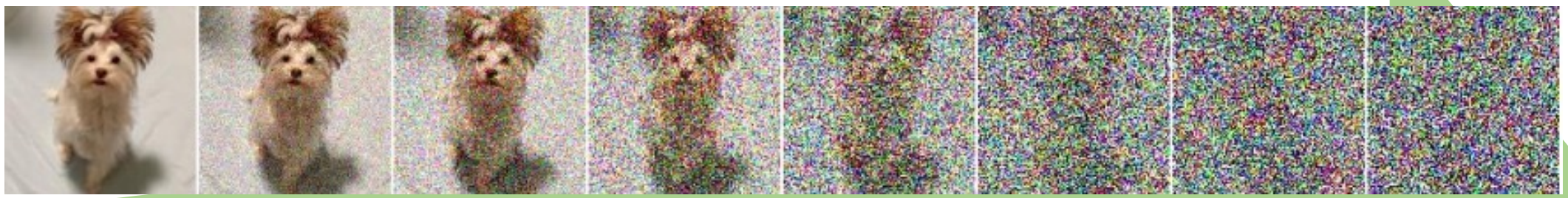


結論

- σ-nice関数を定義できない場合があることを示し、new σ-nice関数を定義した。また、任意の関数がnew σ-nice関数となるための**十分条件**を示した。
- SGDには関数を**平滑化**する効果があり、その度合いは**学習率**と**バッチサイズ**によって決まる。$\delta = \eta\sqrt{\frac{C^2}{b}}$
- この性質を利用した暗黙的な**段階的**最適化アルゴリズムアルゴリズムを提案し、このアルゴリズムが$\mathcal{O}\left(1/\epsilon^{\frac{1}{p}}\right)(p \in (0,1])$ 回の反復で**大域的**最適解$x^\star$のϵ-近傍に到達することを示した。
- したがって、計算資源が十分にあれば、シンプルなSGDでもDNNを含む**非凸**関数の**大域的**最適化が可能となる。

付録：なぜ段階的最適化なのか

- **機械学習分野で既に成功している手法だからです。**
- **スコアベースモデル（2019）と拡散モデル（2020）**
  - **敵対的生成ネットワーク（GANs）を上回る性能を誇る、最先端の画像生成モデルで、暗黙的に段階的最適化のアプローチを利用しています。**

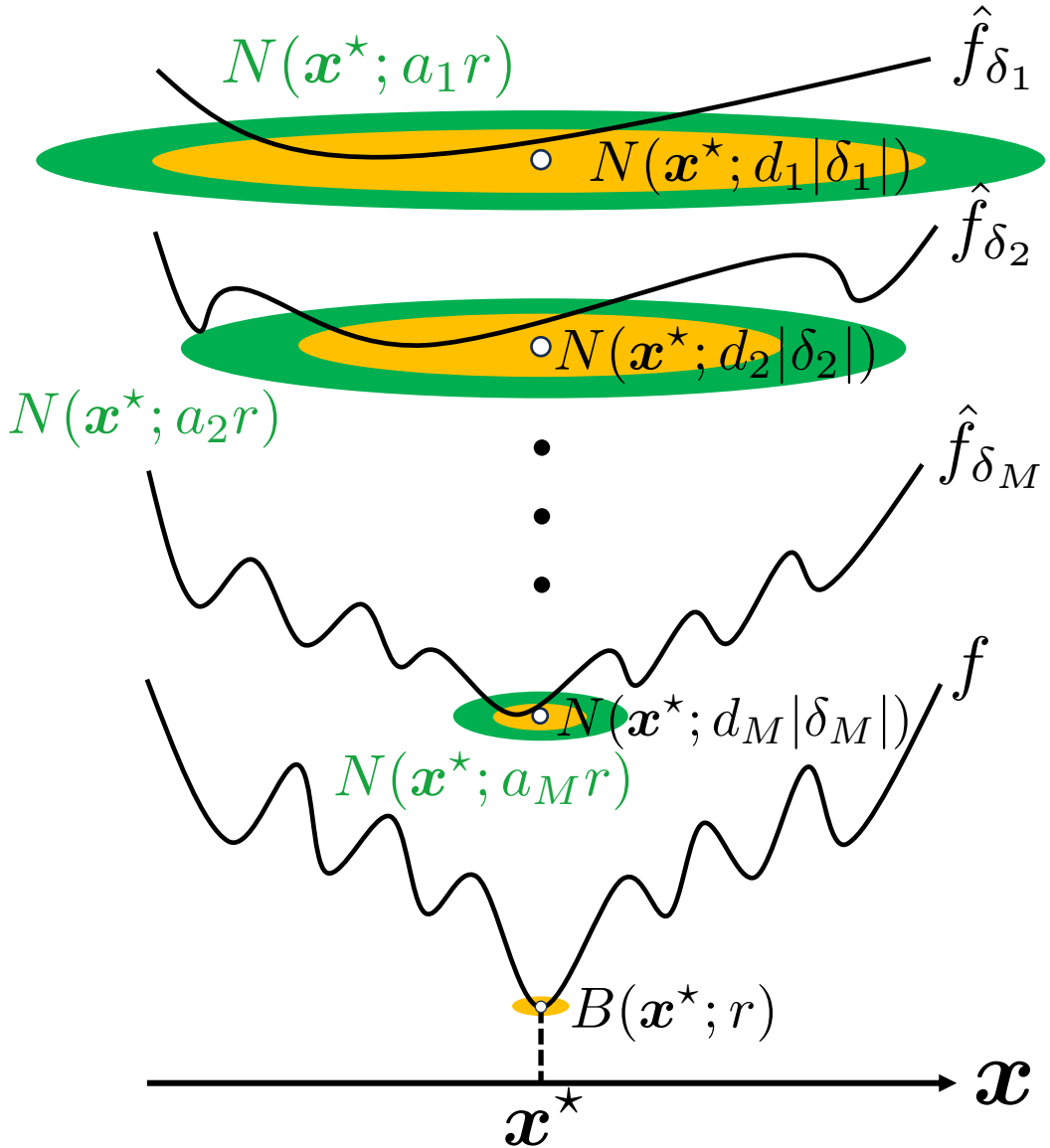


小さいノイズ　　大きいノイズ

➡ **段階的最適化の視点から収束解析を提供できる可能性があります。**

出典：https://yang-song.net/blog/2021/score/

付録：Newσ-nice性の十分条件



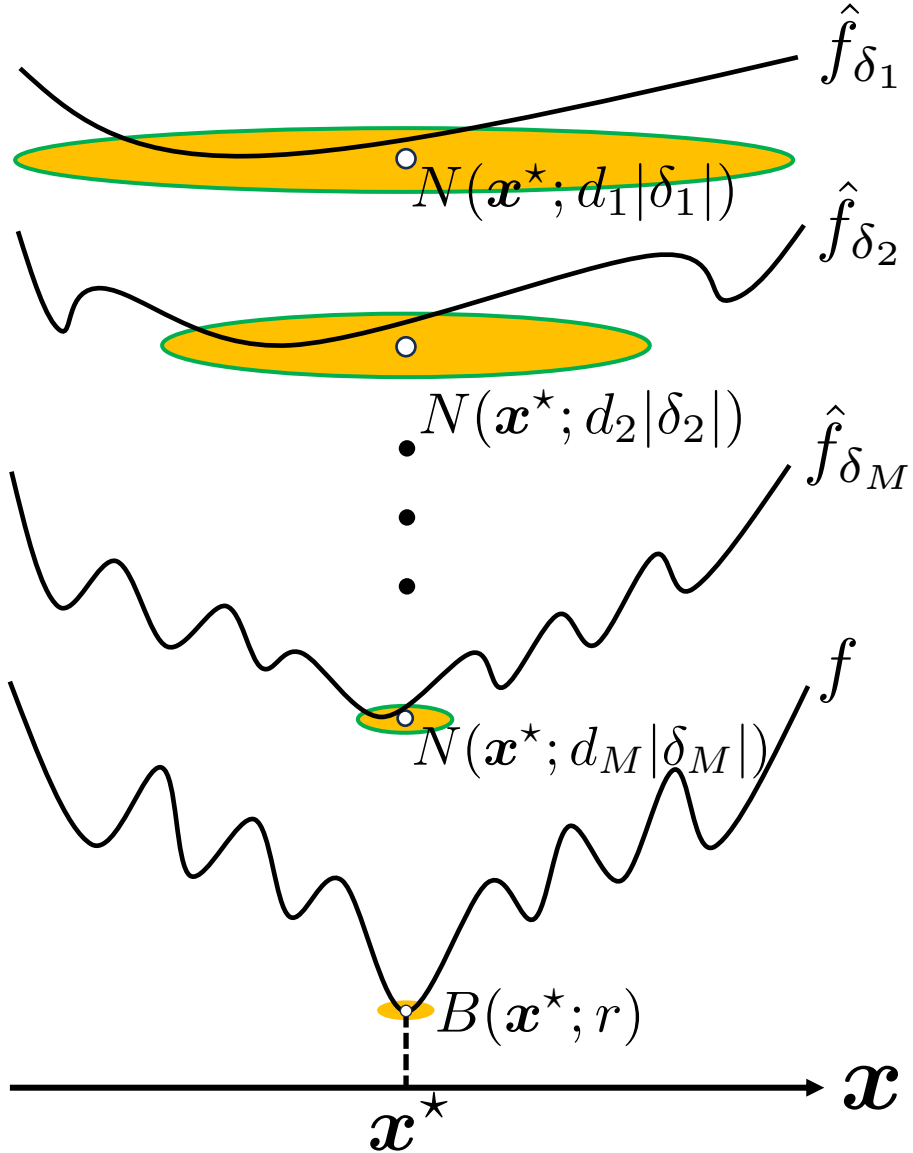
命題3.1

関数$f: \mathbb{R}^d \to \mathbb{R}$が十分小さい$r > 0$に対して閉球$B(\boldsymbol{x}^\star; r)$で$\boldsymbol{\sigma}$–強凸だとし、ノイズレベル$\delta_m$が$|\delta_m| = |\delta^-|$を満たすならば、平滑化された関数$\hat{f}_{\delta_m}$は近傍$N(\boldsymbol{x}^\star; a_m r)(a_m > \sqrt{2})$で$\boldsymbol{\sigma}$-強凸となる。ただし、

$$|\delta^-| := \Big| \|\boldsymbol{x}^\star - \boldsymbol{x}\| \|\boldsymbol{u}_m\| \cos\theta - \sqrt{\|\boldsymbol{x}^\star - \boldsymbol{x}\|^2 \|\boldsymbol{u}_m\|^2 \cos^2\theta - r^2(a_m^2 - 1)} \Big|$$

とし、$\boldsymbol{x} \in N(\boldsymbol{x}^\star; a_m r), \boldsymbol{u}_m \sim B(\boldsymbol{0}; 1)$で、$\theta$は$\boldsymbol{u}_m$と$\boldsymbol{x}^\star - \boldsymbol{x}$のなす角とする。また、$d_m := a_m r / |\delta^-|$とすると、関数$\hat{f}_{\delta_m}$は近傍$N(\boldsymbol{x}^\star; d_m|\delta_m|)$でも$\boldsymbol{\sigma}$-強凸となる。

付録：New σ-nice性の十分条件



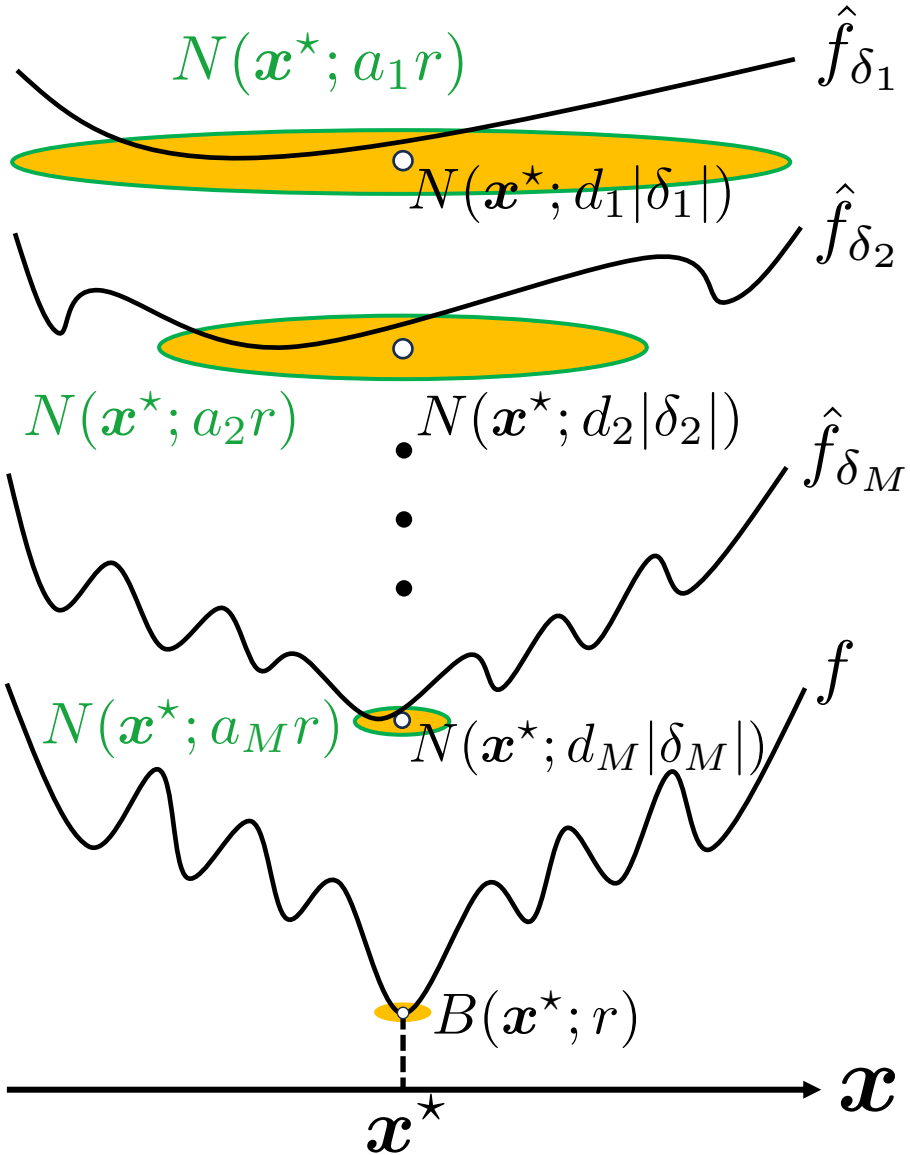
命題3.1

関数$f: \mathbb{R}^d \to \mathbb{R}$が十分小さい$r > 0$に対して閉球$B(\boldsymbol{x}^\star; r)$で$\sigma$–強凸だとし、ノイズレベル$\delta_m$が$|\delta_m| = |\delta^-|$を満たすならば、平滑化された関数$\hat{f}_{\delta_m}$は近傍$N(\boldsymbol{x}^\star; a_m r)(a_m > \sqrt{2})$で$\sigma$-強凸となる。ただし、

$$|\delta^-| := \left| \|\boldsymbol{x}^\star - \boldsymbol{x}\| \|\boldsymbol{u}_m\| \cos\theta - \sqrt{\|\boldsymbol{x}^\star - \boldsymbol{x}\|^2 \|\boldsymbol{u}_m\|^2 \cos^2\theta - r^2(a_m^2 - 1)} \right|$$

とし、$\boldsymbol{x} \in N(\boldsymbol{x}^\star; a_m r), \boldsymbol{u}_m \sim B(\boldsymbol{0}; 1)$で、$\theta$は$\boldsymbol{u}_m$と$\boldsymbol{x}^\star - \boldsymbol{x}$のなす角とする。また、$d_m := a_m r / |\delta^-|$とすると、関数$\hat{f}_{\delta_m}$は近傍$N(\boldsymbol{x}^\star; d_m|\delta_m|)$でも$\sigma$-強凸となる。

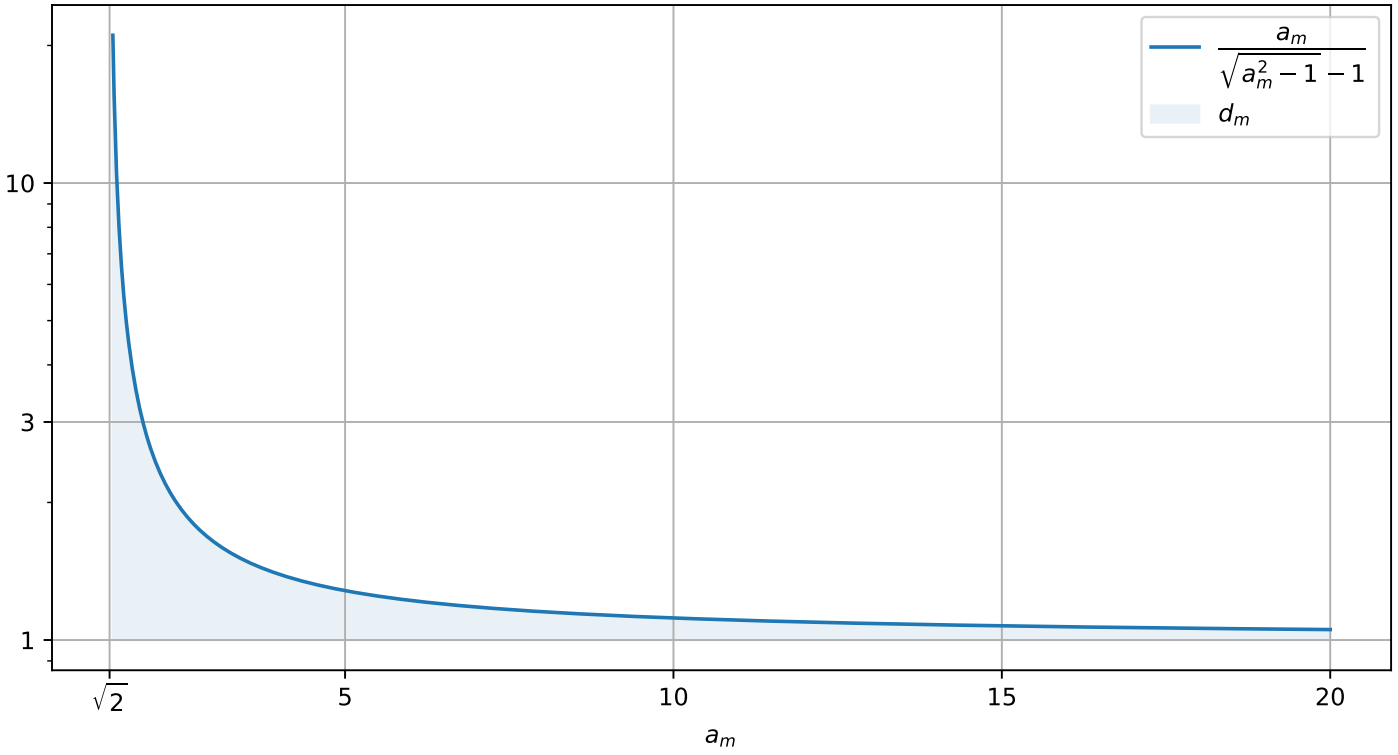
付録：New σ-nice性の十分条件



d_mがとるべき値の範囲

$$d_m := a_m r / |\delta^-|$$

$$\Rightarrow 1 \leq d_m \leq \frac{a_m}{\sqrt{a_m^2 - 1} - 1}$$



付録：σ-nice 関数（再掲）

定義 (σ-nice 関数)

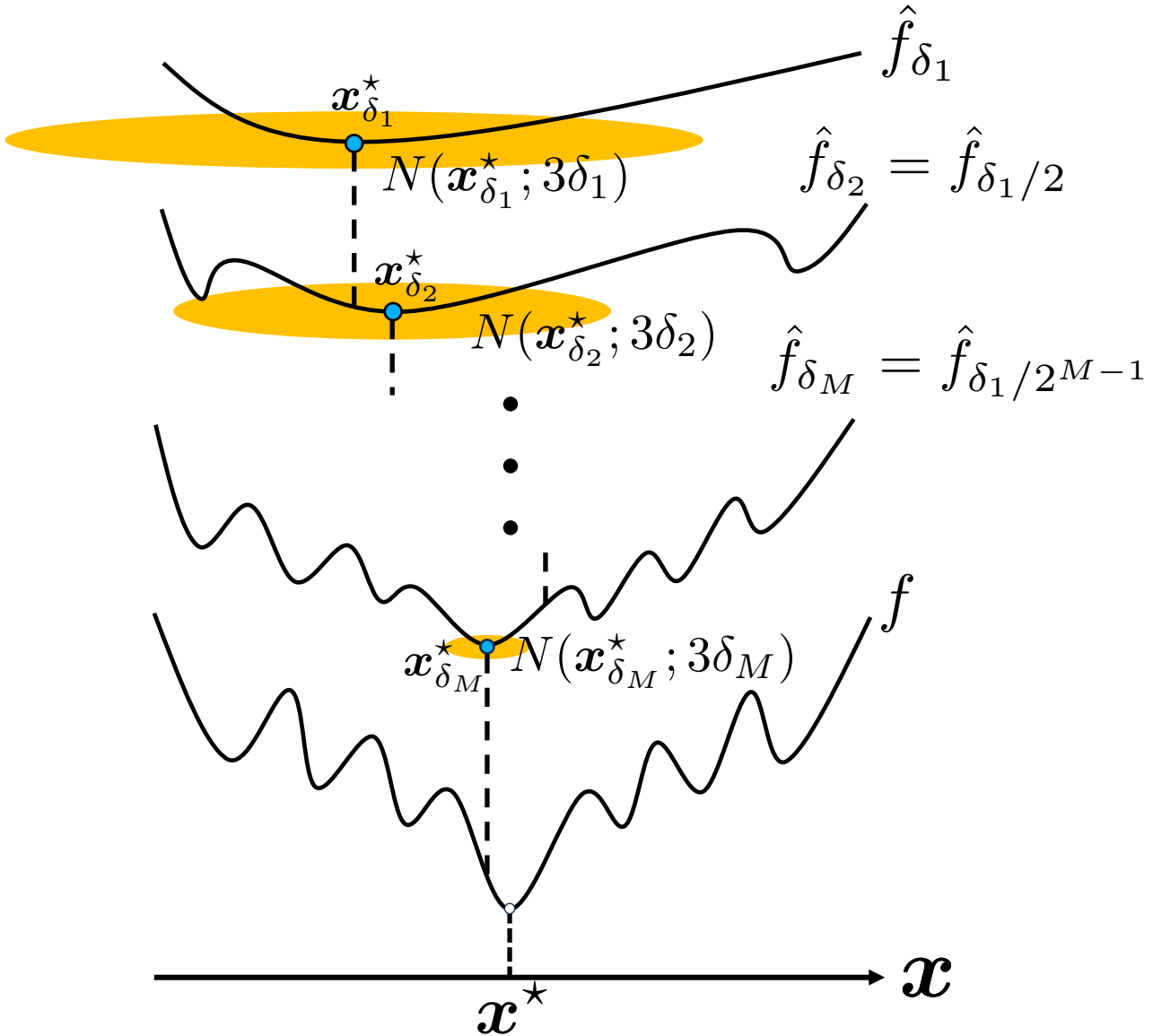
任意の非凸関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f は σ-nice 関数であるという。

（i）任意の $\delta > 0$ と $\boldsymbol{x}^\star_\delta$ に対して、

$$\left\| \boldsymbol{x}^\star_\delta - \boldsymbol{x}^\star_{\delta/2} \right\| \leq \frac{\delta}{2}$$

が成り立つ。

（ii）任意の $\delta > 0$ に対して、関数 $\hat{f}_\delta(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}^\star_\delta; 3\delta)$ で σ–強凸となる。



付録：SGDの挙動についての理論的洞察

▷なぜ大きいバッチサイズは汎化性を悪化させるのか。

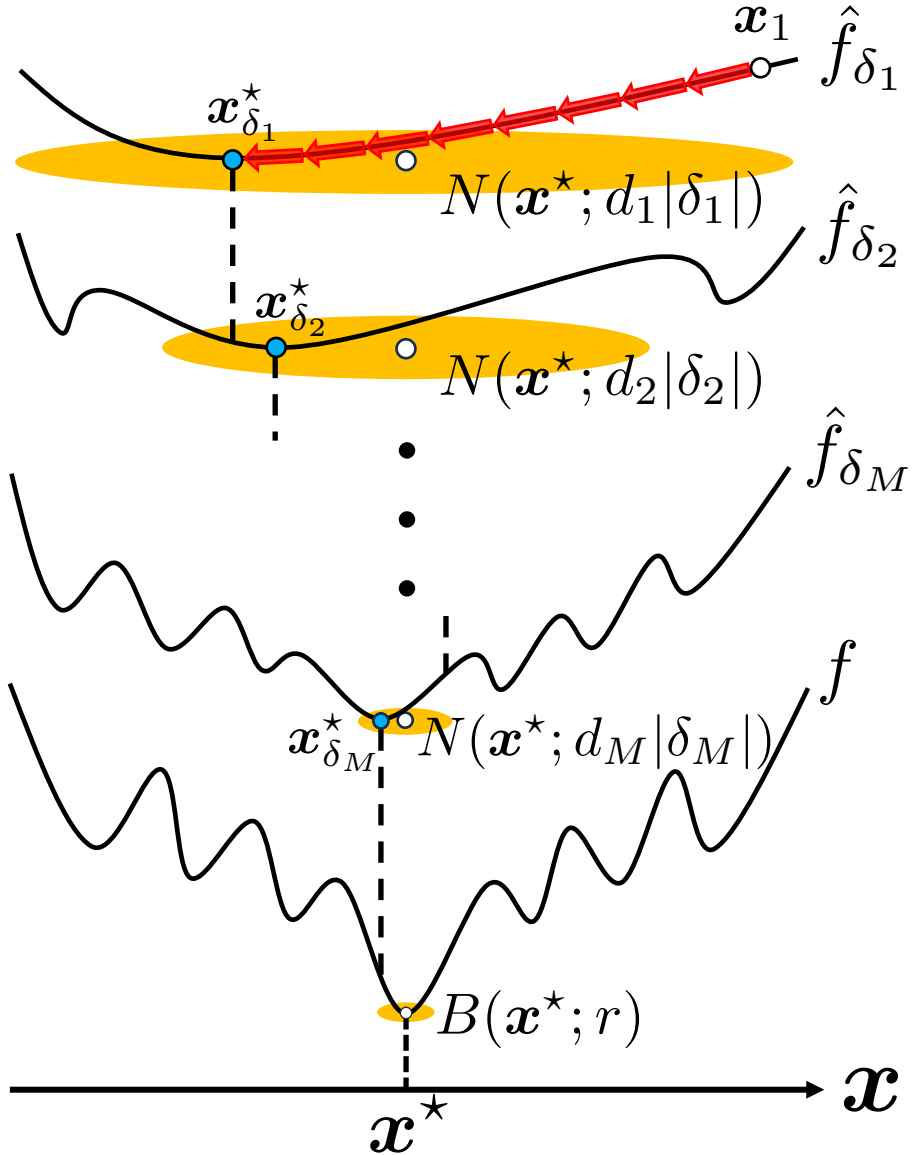
➡ バッチサイズが大きすぎると、最適化される関数は元の非凸関数に近付き、悪い局所的最適解に陥りやすくなるため。

▷なぜ減少学習率または増加バッチサイズは定数よりも優れるのか。

➡ 訓練中に学習率を減少させる、あるいはバッチサイズを増加させることは、ノイズレベルを小さくすることと等価。

➡ 減少学習率または増加バッチサイズを使うことは、まさに段階的最適化のアプローチになっているため、定数よりも優れる。

付録：最適なノイズスケジューリング



命題3.3

関数$f: \mathbb{R}^d \to \mathbb{R}$がnew σ-nice関数ならば、任意の$m \in [M-1]$に対して、

$$\left\| x^\star_{\delta_m} - x^\star \right\| < d_{m+1}|\delta_{m+1}|$$

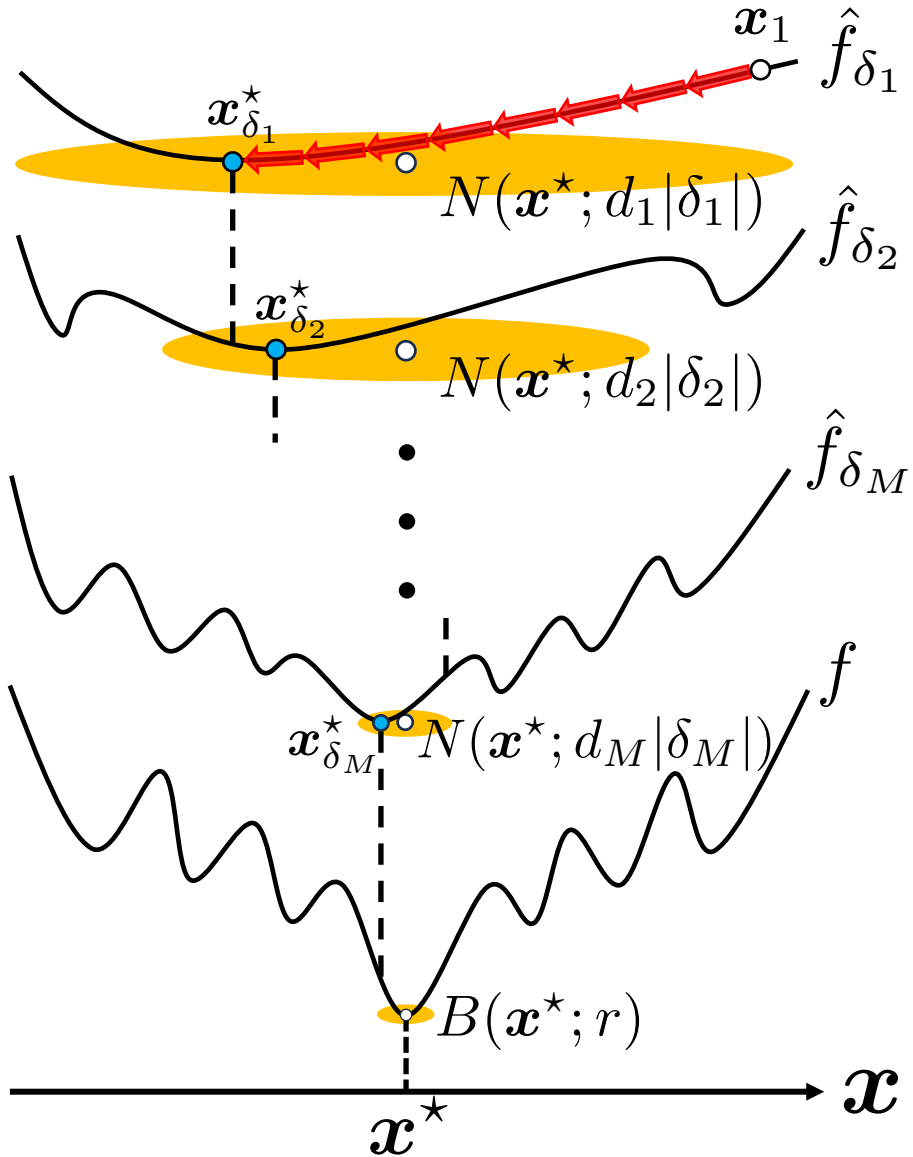
が成り立つ。

- ▷次の関数の最適化の**初期点**が、必ず**強凸領域**に含まれる。
- ▷最初の初期点x_1が関数$\hat{f}_{\delta_1}$の**強凸領域**$N(x^\star; d_1|\delta_1|)$に含まれていれば、大域的最適解$x^\star$に到達できる。
- ▷命題3.3が成り立つためには、

$$\gamma_m \in \left(\frac{1}{d_{m+1}}, 1 \right)$$

が必要。

付録：最適なノイズスケジューリング

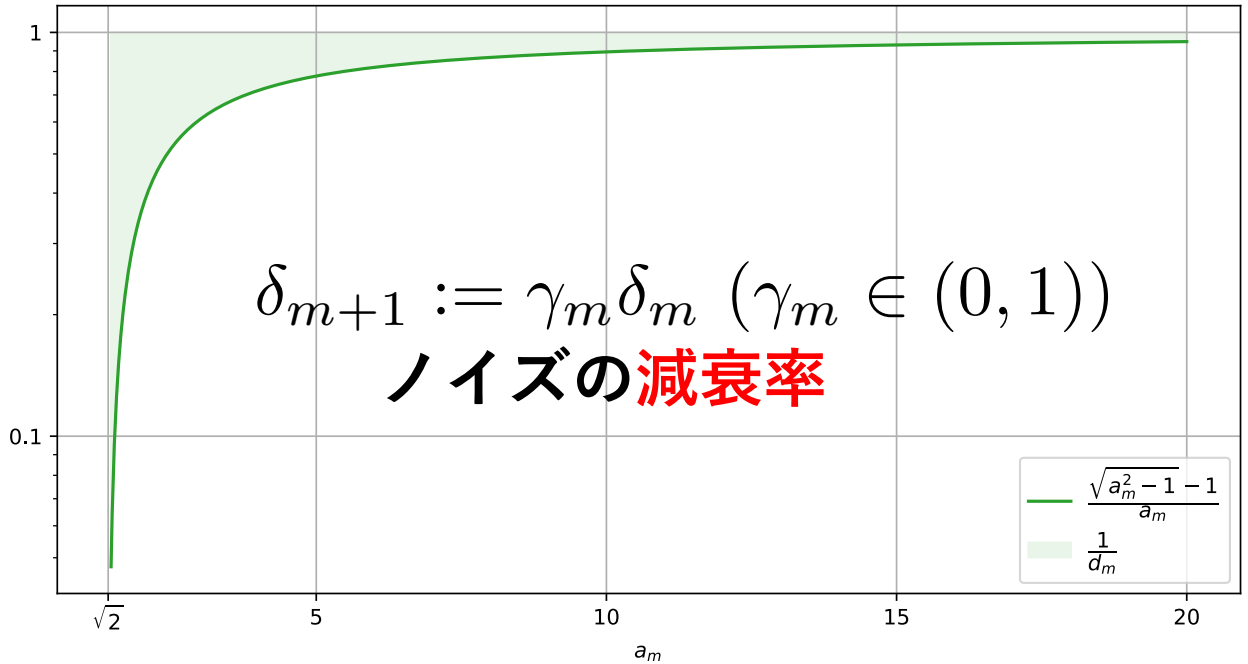


▷**命題3.3が成り立つためには、**

$$\gamma_m \in \left(\frac{1}{d_{m+1}}, 1\right)$$ **が必要。**

$$d_m := a_m r / |\delta^-|$$

$$\Rightarrow \frac{\sqrt{a_m^2 - 1} - 1}{a_m} \leq \frac{1}{d_m} \leq 1$$



$$\delta_{m+1} := \gamma_m \delta_m \ (\gamma_m \in (0, 1))$$

ノイズの減衰率

付録：σ-nice 関数（再掲）

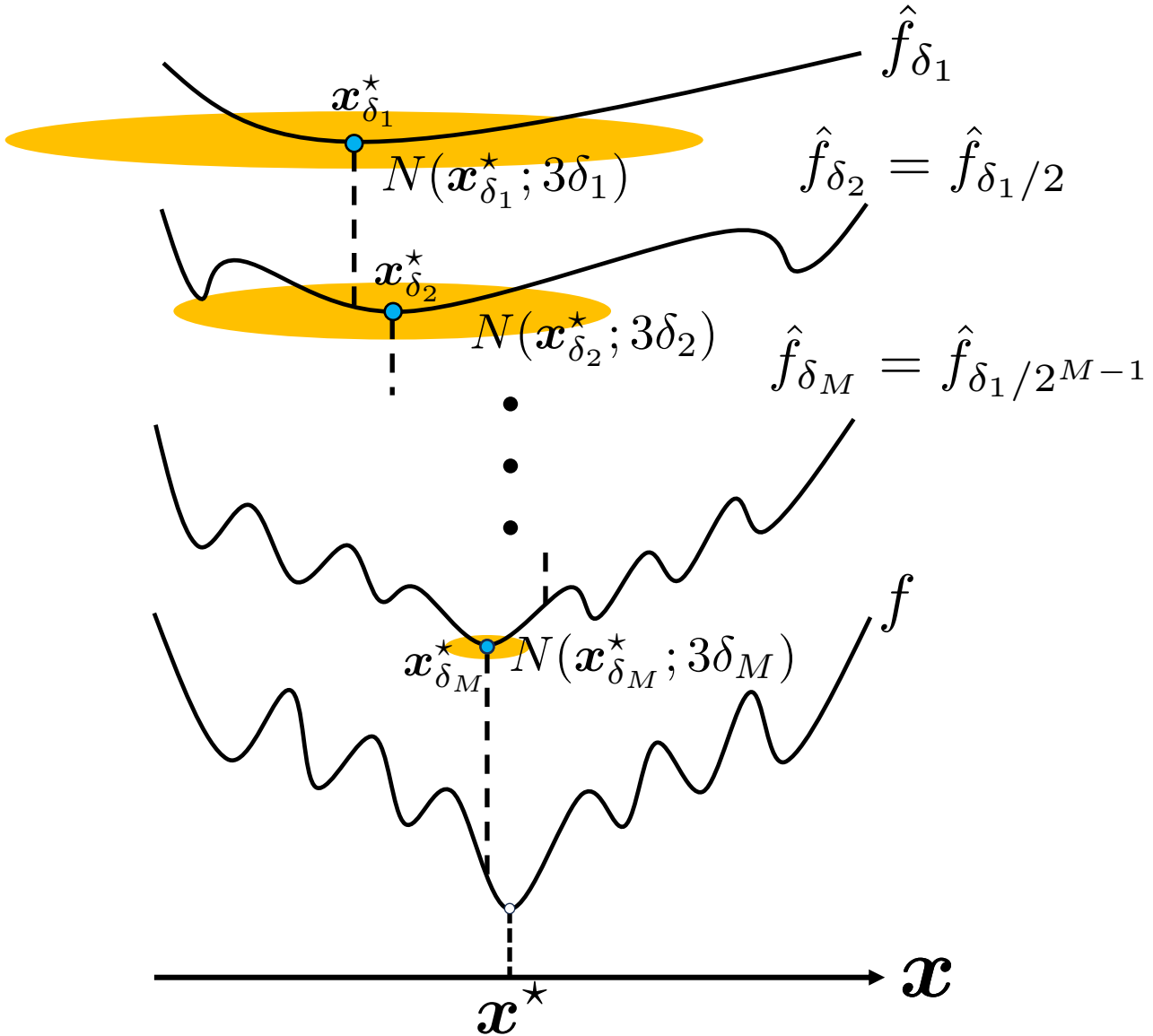
定義 (σ-nice 関数)
任意の非凸関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が次の2つの条件を満たすとき、関数 f は σ-nice 関数であるという。

（i）任意の $\delta > 0$ と $\boldsymbol{x}_\delta^\star$ に対して、

$$\left\| \boldsymbol{x}_\delta^\star - \boldsymbol{x}_{\delta/2}^\star \right\| \leq \frac{\delta}{2} \quad \left(\gamma_m := \frac{1}{2} \right)$$

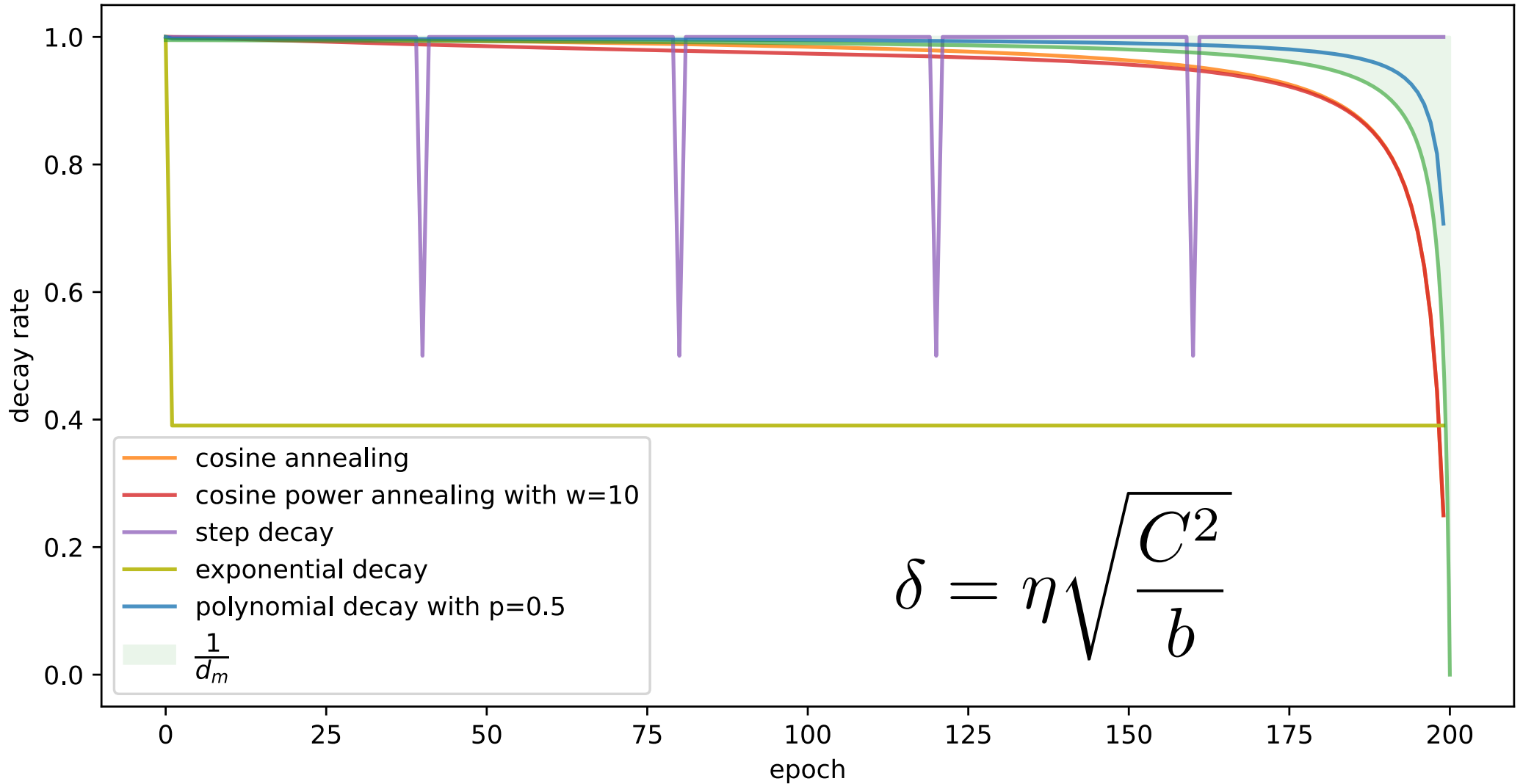
が成り立つ。

（ii）任意の $\delta > 0$ に対して、関数 $\hat{f}_\delta(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}_\delta^\star; 3\delta)$ で σ–強凸となる。

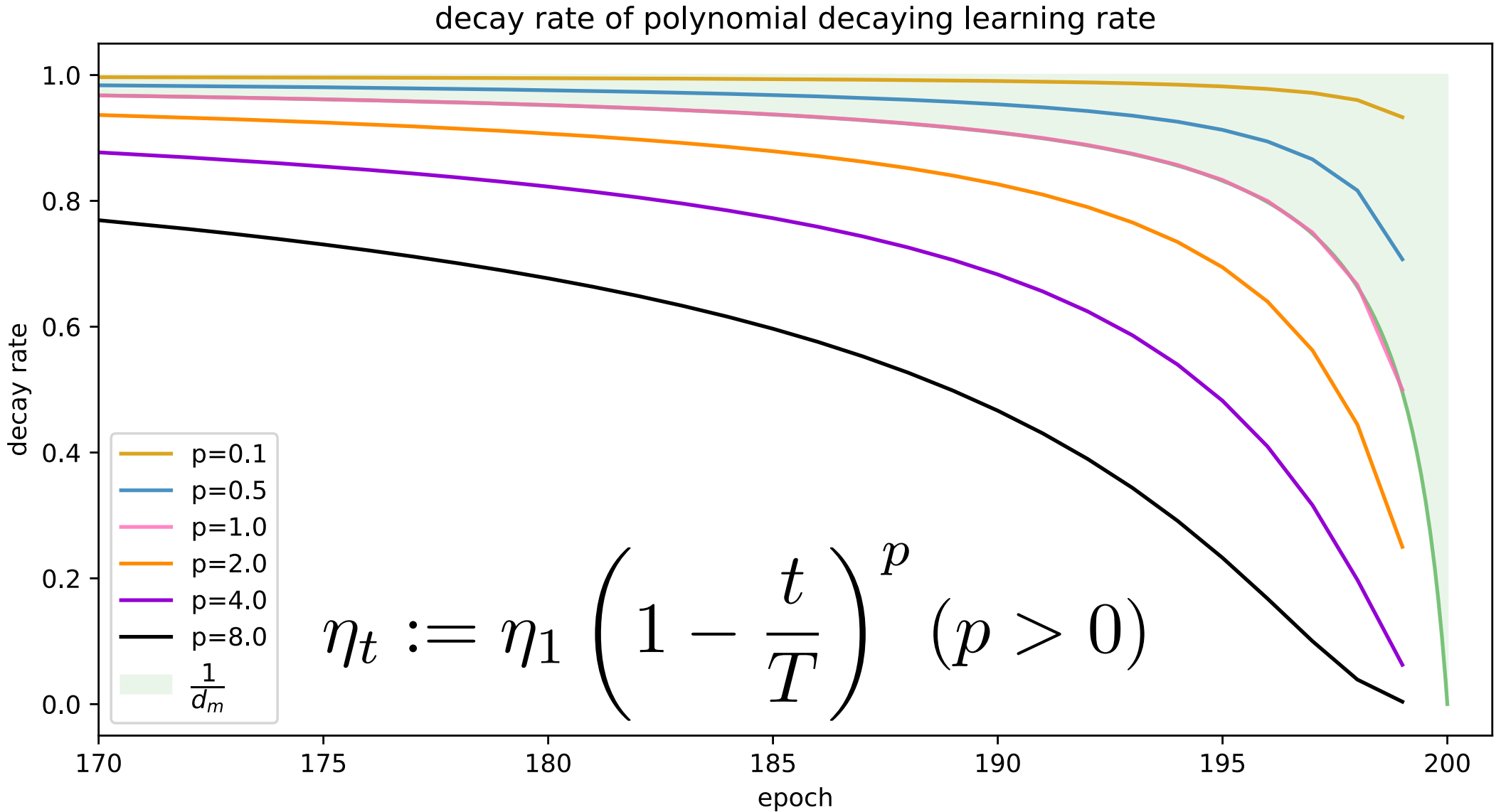


付録：最適な学習率スケジューリング

▷**最適なノイズの減衰率は、直ちに最適な学習率の減衰率となる。**



付録：最適な学習率スケジューリング



付録：暗黙的な段階的最適化アルゴリズム

Algorithm 3 Implicit Graduated Optimization

Require: $\epsilon > 0, p \in (0,1), \bar{d} > 0, \boldsymbol{x}_1, \eta_1, b_1$

$\delta_1 := \frac{\eta_1 C}{\sqrt{b_1}}$

$\alpha_0 := \min\left\{\frac{\sqrt{b_1}}{4L_f\eta_1 C(1+\bar{d})}, \frac{\sqrt{b_1}}{\sqrt{2\sigma}\eta_1 C}\right\}, M^p := \frac{1}{\alpha_0 \epsilon}$

for $m = 1$ to $M+1$ **do**

  if $m \neq M+1$ **then**

    $\epsilon_m := \sigma^2\delta_m^2,\ T_F := H_4/(\epsilon_m - H_3\eta_m)$

    $\gamma_m := \frac{(M-m)^p}{\{M-(m-1)\}^p}$

    $\kappa_m/\sqrt{\lambda_m} = \gamma_m\ (\kappa_m \in (0,1], \lambda_m \geq 1)$

  end if

  $\boldsymbol{x}_{m+1} := \mathrm{GD}(T_F, \boldsymbol{x}_m, \hat{f}_{\delta_m}, \eta_m, b_m)$

  $\eta_{m+1} := \kappa_m\eta_m, b_{m+1} := \lambda_m b_m$

  $\delta_{m+1} := \frac{\eta_{m+1}C}{\sqrt{b_{m+1}}}$

end for

return $\boldsymbol{x}_{M+2}$

定理3.4

関数$f: \mathbb{R}^d \to \mathbb{R}$が$L_f$-リプシッツ new σ-nice 関数だとすると、アルゴリズム3は、$\mathcal{O}\left(1/\epsilon^{\frac{1}{p}}\right)\ (p \in (0,1])$回の反復で、関数$f$の**大域的最適解**$\boldsymbol{x}^\star$の$\epsilon$-近傍に到達する。

Algorithm 4 Gradient Descent (GD)

Require: $T_F, \hat{\boldsymbol{x}}_1, F, \eta, b$

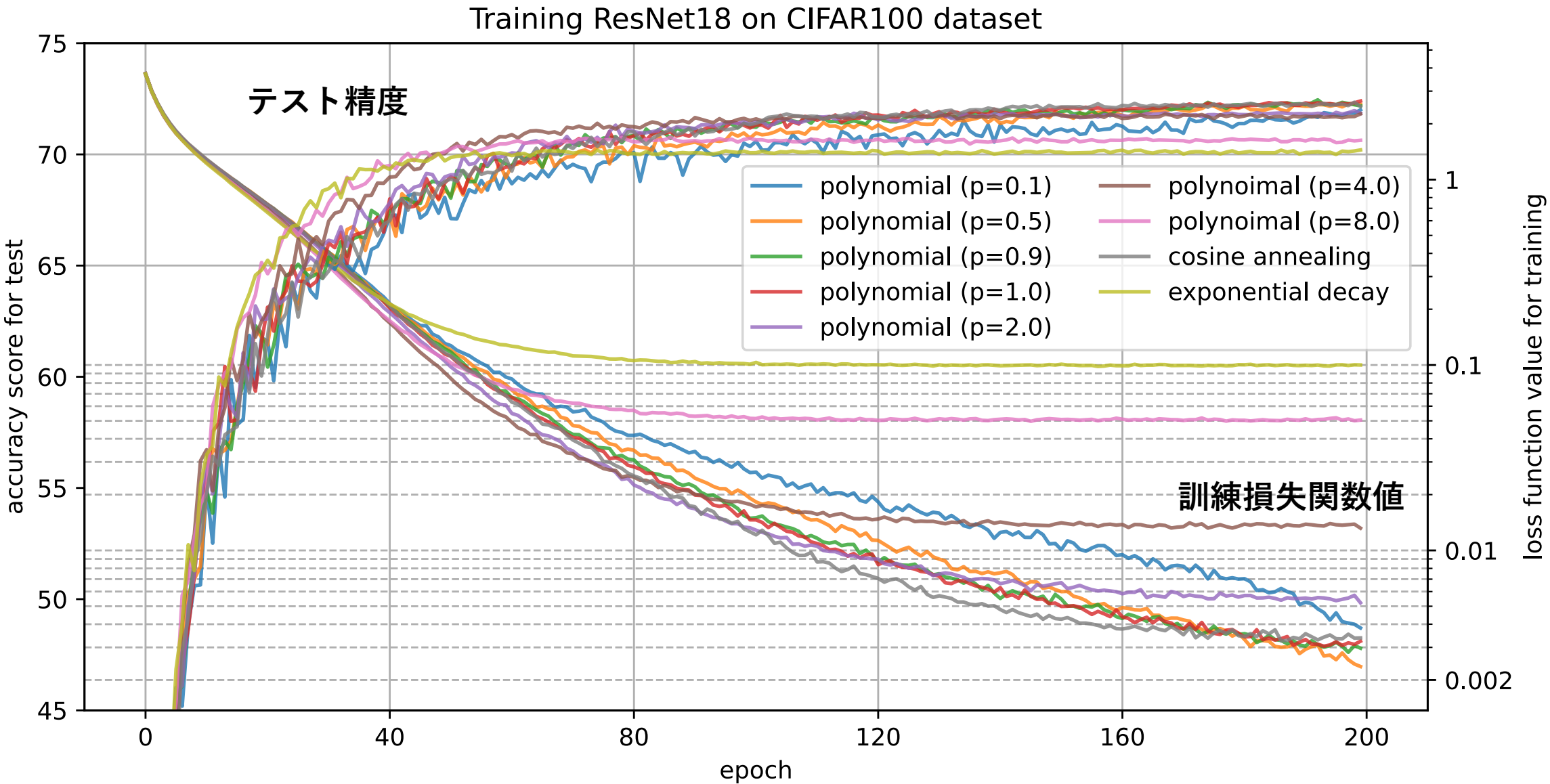
  for $t = 1$ to T_F **do**

    $\hat{\boldsymbol{x}}_{t+1} := \hat{\boldsymbol{x}}_t - \eta\nabla F(\boldsymbol{x}_t)$

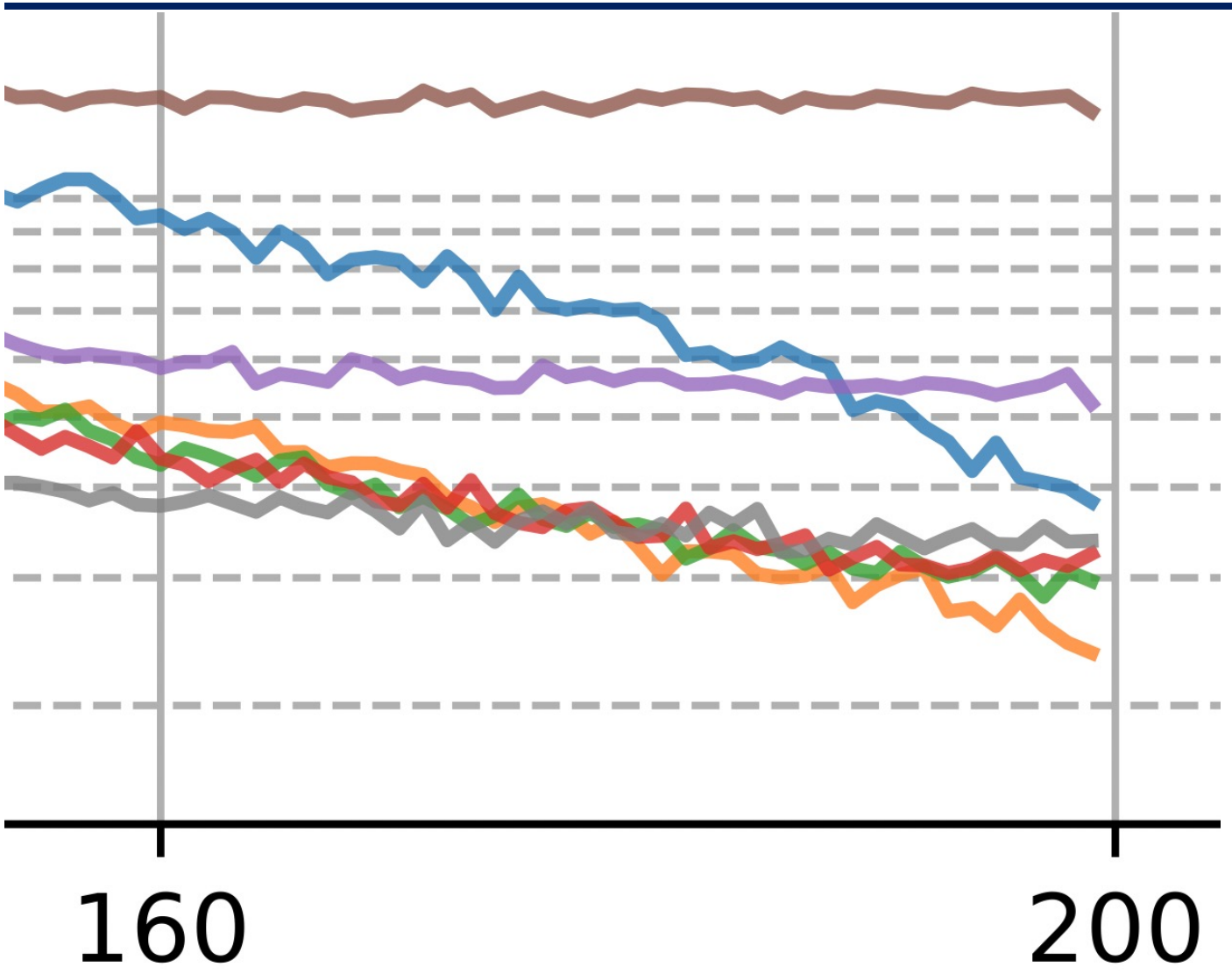
  end for

  return $\hat{\boldsymbol{x}}_{T_F+1} = \mathrm{GD}(T_F, \hat{\boldsymbol{x}}_1, F, \eta)$

付録：数値実験ー最適な学習率の減衰率



付録：数値実験ー最適な学習率の減衰率



$$\mathcal{O}\left(1/\epsilon^{\frac{1}{p}}\right)(p \in (0,1])$$

-- p=4.0
-- p=2.0
◎-- p=0.1
-- コサインアニーリング
◎-- p=1.0
◎-- p=0.9
◎-- p=0.5

◎理論的に最適
(命題3.3)