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DNNの大域的最適化

佐藤　尚樹，飯塚　秀明

近年急速に発展しているテキスト生成をはじめとする機械学習分野において，深層ニューラルネットワーク (DNN) の最適化理論は重要な役割を果たしている．本稿は，深層学習に現れる経験損失最小化問題を解くための最適化法のうち，最もシンプルな確率的勾配降下法について，段階的最適化の観点から考察する．

キーワード：深層学習，非凸最適化，確率的勾配降下法，段階的最適化，平滑化

1. はじめに

訓練データセット $\mathcal{S} := (z_1, z_2, \ldots, z_n)$ と深層ニューラルネットワーク (Deep Neural Networks, DNN) のパラメータ $\boldsymbol{x} \in \mathbb{R}^d$ が与えられたとき，深層学習モデルから正解ラベルとの誤差を表す微分可能な非凸損失関数 $f(\boldsymbol{x}; z_i)$ が得られるとする．このとき，経験損失最小化問題

$$\begin{aligned} &\text{目的関数}: f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) \to \text{最小} \qquad (1)\\ &\text{条件}: \boldsymbol{x} \in \mathbb{R}^d \end{aligned}$$

の最適解を見つけることを，「モデルを訓練する」という．ただし，$f_i(\boldsymbol{x}) := f(\boldsymbol{x}; z_i)$ は訓練データ z_i に対する損失関数とする．このような非凸最適化問題を解くための最急降下法の更新式は，初期点を $\boldsymbol{x}_0 \in \mathbb{R}^d$，学習率を $\eta_t > 0$ $(t \in \mathbb{N})$ とすると，

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

で与えられる．ところが，DNN の次元数 d と訓練データの総数 n は非常に大きいため，一般的な計算機では全勾配 $\nabla f(\boldsymbol{x}_t)$ を計算することができず，最急降下法は実行できない．それに対して，確率的勾配降下法 (Stochastic Gradient Descent, SGD) は，反復回数 t において n 個の訓練データからランダムに選ばれた b $(< n)$ 個のデータ $\mathcal{S}_t$ に対して計算されたミニバッチ確率的勾配

$$\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t) := \frac{1}{b}\sum_{i=1}^{b} \nabla f_{\xi_{t,i}}(\boldsymbol{x}_t)$$

を探索方向に利用するため，バッチサイズ b を適切に設定すれば実行することができる．ただし，確率変数 $\xi_{t,i}$ は反復回数 t での i 番目のデータにより生成される確率変数とする．問題 (1) は無数に局所的最適解をもっていると考えられるが，一般に最急降下法，SGD のいずれも，局所的最適解に収束することは保証できても，大域的最適解に収束することは保証できない．本稿は，問題 (1) の大域的最適解を見つけることを目指す．

2. 数学的準備

$\mathbb{R}^d$ を d 次元ユークリッド空間とし，$\mathbb{R}^d$ のノルムを $\|\cdot\|$ とする．本稿では解析のために，以下のような仮定を認める．

仮定 1. 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，すなわち，

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

が成り立つ．

仮定 2. 関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ は L_f-リプシッツ関数とする，すなわち，

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

が成り立つ．

仮定 3. $(\boldsymbol{x}_t)_{t\in\mathbb{N}}$ を最適化アルゴリズムが生成した点列とすると，

(i) 任意の反復回数 $t \in \mathbb{N}$ に対して，

さとう　なおき，いいづか　ひであき
明治大学大学院理工学研究科
〒 214–8571 神奈川県川崎市多摩区東三田 1–1–1
naoki310303@gmail.com
iiduka@cs.meiji.ac.jp

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

が成り立つ．ただし，$\mathsf{G}_{\xi_t}(\boldsymbol{x}_t)$ は，関数 $f(\cdot)$ の $\boldsymbol{x}_t$ における確率的勾配であり，$\mathbb{E}_{\xi_t}[\cdot]$ は確率変数 ξ_t に関する期待値である．

(ii) 非負定数 $C^2 \geq 0$ が存在して，任意の反復回数 $t \in \mathbb{N}$ に対して，

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

が成り立つ．

仮定 4. 任意の反復回数 $t \in \mathbb{N}$ において，全勾配 ∇f は，ミニバッチ $\mathcal{S}_t \subset \mathcal{S}$ で次のように近似される．

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

仮定 3 (i) は，各反復回数 t で得られる f の確率的勾配 G_{ξ_t} の期待値が全勾配 ∇f と一致することを意味し，仮定 3 (ii) は，確率的勾配 G_{ξ_t} と全勾配 ∇f の二乗ノルムの意味での差の期待値が C^2 以下であることを意味している．すなわち，C^2 は確率的勾配 G_{ξ_t} の分散の上界を表している．仮定 4 は，SGD の探索方向であるミニバッチ確率的勾配 $\nabla f_{\mathcal{S}_t}(\boldsymbol{x}_t)$ が，b 個の確率的勾配 $\mathsf{G}_{\xi_{t,i}}(i \in [b])$ の平均で計算されることを意味している．ただし，$[b] := \{1, 2, \cdots, b\}$ とする．

3. 段階的最適化

段階的最適化 (Graduated Optimization) は Blake と Zisserman によって提案された，非凸最適化問題の大域的最適解を探索する大域的最適化手法の一つである [1]．まず徐々に小さくなるノイズ $(\delta_m)_{m\in[M]}$ による平滑化演算によって，徐々に元の目的関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ に近づくように平滑化された M 個の関数の列 $(\hat{f}_{\delta_m})_{m\in[M]}$ を用意する．そして，最も平滑化された関数 $\hat{f}_{\delta_1}$ を最初に最適化し，その近似解を初期点として 2 番目に大きく平滑化された関数 $\hat{f}_{\delta_2}$ を最適化し，次に 2 番目の近似解を初期点として 3 番目に大きく平滑化された関数 $\hat{f}_{\delta_3}$ を最適化する，という手順を繰り返すことで，元の目的関数 f の局所的最適解を避けて大域的最適解を探索する．段階的最適化の概念図については，文献 [2](Figure 1) を参照されたい．一般に関数の平滑化は，正規分布や一様分布に従う確率変数で関数を畳み込むことで実現される．本稿では解析の都合上，平滑化に使用される確率変数は一様分布に従うとする．

定義 1 (関数の平滑化)**.** L_f-リプシッツ関数 f を平滑化して得られる関数 $\hat{f}_\delta\colon \mathbb{R}^d \to \mathbb{R}$ は，

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

と表される．ここで，$\delta \in \mathbb{R}$ は平滑化の度合いを表し，$\boldsymbol{u} \in \mathbb{R}^d$ は閉球 $B(\mathbf{0};1)$ から一様にサンプリングされたベクトルである．また，

$$\boldsymbol{x}^\star := \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}),\quad \boldsymbol{x}_\delta^\star := \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \hat{f}_\delta(\boldsymbol{x})$$

とする．

段階的最適化手法はやノイズ除去 [3] やロバスト推定 [4] など，画像処理分野や機械学習分野で広く利用されている．特に，現在最先端の性能を有する生成モデルであるスコアベースモデル [5] と拡散モデル [6, 7] は，暗黙的に段階的最適化を利用している．その一方で，理論的な研究は少ない．段階的最適化の理論と応用に関する包括的な調査については，文献 [8] を参照されたい．

Hazan らは，段階的最適化アルゴリズムが大域的最適解に収束するような条件を満たす，特別な非凸関数である，σ-nice 関数を定義した [9].

定義 2 (σ-nice 関数)**.** 任意の関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ に対して，次の二つの条件が満たされるとき，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，

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

を満たすような $\boldsymbol{x}_{\delta/2}^\star$ が存在する．

(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-強凸である．

Hazan らは σ-nice 関数に対して以下のような段階的最適化アルゴリズムを使用すると，$\mathcal{O}\left(1/\epsilon^2\right)$ 回の反復で目的関数 f の大域的最適解 $\boldsymbol{x}^\star$ の ϵ-近傍に到達できることを示した [9](Theorem 4.1).

Require: $\sigma > 0, \delta_1 > 0, M \in \mathbb{N}, \boldsymbol{x}_1 \in \mathbb{R}^d$

1: **for** $m = 1$ to $M + 1$ **do**
2: $\quad T_F := \sigma\delta_m^2/32$
3: $\quad \boldsymbol{x}_{m+1} := \mathrm{SGD}(T_F, \boldsymbol{x}_m, \hat{f}_{\delta_m})$
4: $\quad \delta_{m+1} := \delta_m/2$

5: **end for**
6: **Return:** $\boldsymbol{x}_{M+2}$

ところが，一般に DNN を含む複雑で巨大な関数に定義 1 の平滑化演算を施すことはできないため，上記のアルゴリズムによる最適化は実現できない．

4. 確率的ノイズによる平滑化

SGD の探索方向と最急降下方向との間には，各反復で，SGD が一度にすべてのデータを扱えないために

$$\boldsymbol{\omega}_t := \nabla f_{S_t}(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_t)$$

だけ確率的ノイズが生じている．凸最適化における SGD は，収束するために最急降下法よりも多くの反復回数を要する代わりに，1 反復あたりの計算量は少なくなる [10] ことが知られている．すなわち，凸最適化では計算量を減らすためだけにミニバッチ化するのであって，確率的ノイズは邪魔な副産物でしかない．一方，非凸最適化においては，このノイズは非常に重要で，局所的最適解からの脱出を助け [11]，収束は遅いが最急降下法よりも良い汎化性をもたらす [12] ことが経験的に知られている．さらに，Kleinberg らは，以下のような議論から，確率的ノイズ $\boldsymbol{\omega}_t$ が目的関数を平滑化している可能性を指摘した [13]．

反復回数 t において，最急降下法で点列を更新した先を $\boldsymbol{y}_t$，SGD で点列を更新した先を $\boldsymbol{x}_{t+1}$ とする，すなわち，

$$\begin{aligned}\boldsymbol{y}_t &:= \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t),\\ \boldsymbol{x}_{t+1} &:= \boldsymbol{x}_t - \eta \nabla f_{S_t}(\boldsymbol{x}_t)\end{aligned}$$

とすると，

$$\mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_{t+1}\right] = \mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_t\right] - \eta \nabla \mathbb{E}_{\boldsymbol{\omega}_t}\left[f(\boldsymbol{y}_t - \eta\boldsymbol{\omega}_t)\right] \quad (2)$$

が成り立つ．式 (2) の導出は，文献 [2] の A 章を参照されたい．新たに関数 $\hat{f}(\boldsymbol{y}) := \mathbb{E}_{\boldsymbol{\omega}_t}\left[f(\boldsymbol{y} - \eta\boldsymbol{\omega}_t)\right]$ を定義すれば，

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

が成り立つことから，SGD で関数 $f(\boldsymbol{x})$ を最適化することと，最急降下法で関数 $\hat{f}(\boldsymbol{y})$ を最適化することは，期待値の意味では等価である．さらに $\boldsymbol{\omega}_t$ は確率変数であるから，定義 1 より，関数 $\hat{f}(\boldsymbol{y})$ は関数 $f(\boldsymbol{x})$ をある程度平滑化した関数だといえる．

5. SGD の平滑化特性

それでは，4 節の議論をより深め，SGD の確率的ノイズ $\boldsymbol{\omega}_t$ による平滑化の度合い δ は何によって定まるのかを考察する．

仮定 3 (ii) と仮定 4 が成り立つとすると，

$$\mathbb{E}_{\xi_t}\left[\|\boldsymbol{\omega}_t\|\right] \leq \frac{C}{\sqrt{b}} \quad (3)$$

が成り立つ．証明は，文献 [2] の B 章を参照されたい．不等式 (3) を満たすような $\boldsymbol{\omega}_t$ は，正規分布に従う確率変数 $\boldsymbol{u}_t \sim \mathcal{N}\left(\boldsymbol{0}, \frac{1}{\sqrt{d}} I_d\right)$ を使って，

$$\boldsymbol{\omega}_t = \frac{C}{\sqrt{b}} \boldsymbol{u}_t$$

と表すことができる．ただし，I_d は d 次元の単位行列とする．確率的ノイズが正規分布に従うことについては，文献 [14] を参照されたい．これを利用して式 (2) をさらに変形すると，

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_{t+1}\right]\\ &= \mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_t\right] - \eta \nabla \mathbb{E}_{\boldsymbol{\omega}_t}\left[f(\boldsymbol{y}_t - \eta\boldsymbol{\omega}_t)\right]\\ &= \mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_t\right] - \eta \nabla \mathbb{E}_{\boldsymbol{u}_t \sim \mathcal{N}\left(\boldsymbol{0}; \frac{1}{\sqrt{d}} I_d\right)}\\ &\quad \left[f\left(\boldsymbol{y}_t - \frac{\eta C}{\sqrt{b}}\boldsymbol{u}_t\right)\right]\\ &\approx \mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_t\right] - \eta \nabla \mathbb{E}_{\boldsymbol{u}_t \sim B(\boldsymbol{0};1)}\left[f\left(\boldsymbol{y}_t - \frac{\eta C}{\sqrt{b}}\boldsymbol{u}_t\right)\right]\\ &= \mathbb{E}_{\boldsymbol{\omega}_t}\left[\boldsymbol{y}_t\right] - \eta \nabla \hat{f}_{\frac{\eta C}{\sqrt{b}}}(\boldsymbol{y}_t) \quad (4)\end{aligned}$$

が成り立つ．ただし，式 (4) の導出では，次元 d が十分大きいとき，標準正規分布は半径 $\sqrt{d}$ の球体上の一様分布に近似できる [15] ことを利用している．式 (4) は，SGD で関数 $f(\boldsymbol{x})$ を最適化することと，最急降下法で関数 $\hat{f}_{\frac{\eta C}{\sqrt{b}}}(\boldsymbol{y})$ を最適化することは，期待値の意味では等価であることを意味している．また，SGD の確率的ノイズ $\boldsymbol{\omega}_t$ による平滑化の度合いは，

$$\delta = \frac{\eta C}{\sqrt{b}} \quad (5)$$

となり，学習率 η，バッチサイズ b，確率的勾配の分散 C^2 によって定まるといえる．すなわち，学習率が大きく，バッチサイズが小さいほど平滑化の度合いは大きくなる．この発見によって，これまで実験的には観察されていたが，理論的には説明できなかった多くの現象を説明することができる．

6. SGD の挙動についての理論的洞察

Keskar らは，SGD でモデルを訓練するとき，大きいバッチサイズを使うとその近傍が急峻な局所的最適解

に陥り，汎化性が損なわれることを実験で示した [12]．この現象は経験的にはよく知られており，バッチサイズが大きくても汎化性を損なわないようにするための手法がいくつか提案されている [16, 17]．この現象については，式 (5) から，バッチサイズが大きいときは目的関数 f の平滑化の度合いは小さく，最急降下法によって最適化されるとみなせる関数 $\hat{f}_{\frac{\eta C}{\sqrt{b}}}$ は元の非凸関数 f に近くなるため，その近傍が急峻な局所的最適解に陥りやすいといえる．一方，バッチサイズが小さいときは，目的関数は十分に平滑化されているために，元の非凸関数 f の急峻な局所的最適解は消失し，SGD で更新される点列は汎化性の優れる解に収束するといえる．

多くの先行研究が，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と，定数学習率と定数バッチサイズを使う場合よりも訓練損失関数値，テスト精度の両方で性能が優れることを示している [18, 19]．式 (5) によると，学習率 η を減少させる，あるいはバッチサイズ b を増加させることは平滑化の度合い δ を減少させることと等価である．したがって，減少学習率または増加バッチサイズを使用することは，まさに段階的最適化のアプローチそのものであり，これらは暗黙のうちに近傍が急峻な局所的最適解を避けることに寄与していたといえる．

上記の考察は，その近傍が急峻な局所的最適解よりも，近傍が平坦な局所的最適解の方が汎化性に優れるという仮説 [12, 20–22] に基づいていることに注意したい．訓練データは限られているため，訓練データのみによって構成される訓練損失関数と，テストデータを含む未知のデータによって構成される未知の関数の間には誤差があるはずである．近傍が平坦な局所的最適解ならば，その誤差の影響を受けにくいために高い汎化性を担保できる，という直感的な説明は可能だが，厳密に理論的に証明されたことはない．

7. new σ-nice 関数

Hazan らが提案した σ-nice 関数 [9] は，どの程度特別な関数なのか不明であった．そこで，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 関数であるための十分条件を示す．

定義 3 (new σ-nice 関数)**.** $\delta_1 \in \mathbb{R}$ とする．任意の関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ に対して次の二つの条件が満たされるとき，関数 f は new σ-nice 関数であるという．

(i) 任意の $m \in [M]$,$\gamma_m \in (0,1)$ に対して $|\delta_{m+1}| := \gamma_m|\delta_m|$ を満たす $\delta_m \in \mathbb{R}$ と $\boldsymbol{x}^\star_{\delta_m}$ が存在して，

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

が成り立つ．

(ii) 任意の $m \in [M]$,$\gamma_m \in (0,1)$,$|\delta_{m+1}| := \gamma_m|\delta_m|$ を満たす $\delta_m \in \mathbb{R}$ と $d_m > 1$ が存在して，関数 $\hat{f}_{\delta_m}(\boldsymbol{x})$ は近傍 $N(\boldsymbol{x}^\star; d_m\delta_m)$ で σ-強凸である．

定義 3 において，γ_m はノイズ δ_m から δ_{m+1} への減衰率を表しており，σ-nice 関数の定義では m によらず $\gamma_m = 0.5$ であった．また，徐々に狭まる強凸領域の中心が，σ-nice 関数の定義では平滑化後の関数 $\hat{f}$ の大域的最適解 $\boldsymbol{x}^\star_{\delta_m}$ であるのに対して，定義 3 では元の目的関数 f の大域的最適解 $\boldsymbol{x}^\star$ であり，その半径は σ-nice 関数の定義では常に 3δ であるのに対して，定義 3 では $d_m\delta_m$ であることに注意する．

次の二つの命題は，関数 f が new σ-nice 関数であるための十分条件を示している．証明は文献 [2] の D.5 節と D.6 節を参照されたい．

命題 1. 関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が，十分小さな正数 $r > 0$ に対して閉球 $B(\boldsymbol{x}^\star; r)$ で σ-強凸で，ノイズ δ_m が $|\delta_m| = |\delta_m^-|$ を満たすとする．このとき，関数 $\hat{f}_{\delta_m}$ は近傍 $N(\boldsymbol{x}^\star; a_m r)$ で σ-強凸となる ($a_m > \sqrt{2}$)．ただし，$\boldsymbol{x} \in N(\boldsymbol{x}^\star; a_m r), \boldsymbol{u}_m \sim B(\boldsymbol{0}; 1)$ に対して

$$|\delta_m^-| := \sup_{\boldsymbol{x} \in N(\boldsymbol{x}^\star; a_m r)\setminus\{\boldsymbol{x}^\star\}} \mathbb{E}_{\boldsymbol{u}_m \sim B(\boldsymbol{0};1)}\left[\left|z - \sqrt{z^2 - r^2(a_m^2 - 1)}\right|\right]$$

とし，$z := \|\boldsymbol{x}^\star - \boldsymbol{x}\|\|\boldsymbol{u}_m\|\cos\theta$ で，θ は $\boldsymbol{u}_m$ と $\boldsymbol{x}^\star - \boldsymbol{x}$ のなす角とする．また，$d_m := a_m r/|\delta_m^-|$ が成り立つとき，平滑化された関数 $\hat{f}_{\delta_m}$ は近傍 $N(\boldsymbol{x}^\star; d_m|\delta_m|)$ で σ-強凸となる．

命題 1 から，定義 3 の条件 (i) を満たすためには，$d_m := a_m r/|\delta_m^-|$ が成り立つ必要がある．δ_m^- の定義から，d_m のとりうる値の範囲は，

$$1 \le d_m \le \frac{a_m}{\sqrt{a_m^2 - 1} - 1} \tag{6}$$

であることがわかる．図 1 は d_m がとりうる値の範囲を塗りつぶしたものである．σ-nice 関数の定義においては，強凸領域の半径は m によらず $d_m = 3$ であったが，これは a_m が大きいとき，すなわちノイズ $|\delta_m|$ が大きいときには成り立たないことがわかる．

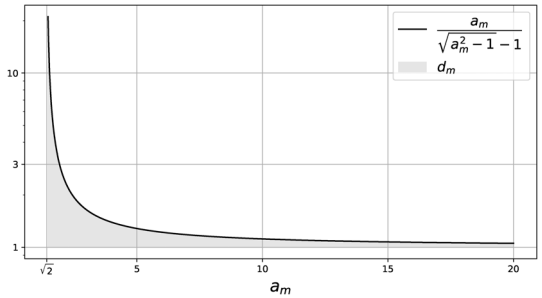

図 1　d_m がとりうる値の範囲

命題 2.　関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ が，十分小さな正数 $r > 0$ に対して閉球 $B(\boldsymbol{x}^\star; r)$ で σ-強凸で，任意の $m \in [M]$ に対して，$\boldsymbol{x}^\star_{\delta_m}$ は近傍 $N(\boldsymbol{x}^\star; d_m|\delta_m|)$ 内に含まれると仮定する．ただし，$d_m > 1, \delta_m \in \mathbb{R}$ とする．このとき，関数 f が new σ-nice 関数であるための十分条件は，任意の $m \in [M]$ に対して，ノイズの大きさ $|\delta_m|$ が次の条件を満たすことである．

$$\frac{2L_g \max\left\{\left\|\boldsymbol{x}^\star_{\delta_m} - \boldsymbol{x}^\star\right\|, \left\|\boldsymbol{x}^\star_{\delta_{m+1}} - \boldsymbol{x}^\star\right\|\right\}}{\sigma\left(1-\gamma_m\right)} \leq |\delta_m| = \left|\delta_m^-\right|. \tag{7}$$

命題 2 より，式 (7) を満たすようなノイズ δ_m で関数を平滑化すれば，任意の関数 f は new σ-nice 関数となる．すなわち，ある関数が new σ-nice 関数であるためには特別な仮定は必要なく，適切な大きさのノイズで平滑化することが唯一の条件となる．

8.　最適なノイズスケジューリング

new σ-nice 関数に対して段階的最適化アルゴリズムを適用するとき，関数 $\hat{f}_{\delta_{m-1}}$ を適当な最適化手法で最適化して得られた近似解が，次の関数 $\hat{f}_{\delta_m}$ の最適化の初期点となる．このとき，関数 $\hat{f}_{\delta_m}$ の強凸領域は関数 $\hat{f}_{\delta_{m-1}}$ の強凸領域よりも狭いため，関数 $\hat{f}_{\delta_m}$ の最適化の出発点が，関数 $\hat{f}_{\delta_m}$ の強凸領域に含まれる保証はない．また，平滑化された関数の最適解がその関数の強凸領域に含まれなければ，最急降下法の点列が強凸領域に留まる保証はない．次の命題はこれらを保証するもので，段階的最適化アルゴリズムによる最適化の成功に不可欠なものである．証明は文献 [2] の D.7 節を参照されたい．

命題 3.　任意の $m \in [M]$ に対して $d_m > 1$ とし，関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ は new σ-nice 関数であるとする．このとき，

(i) 任意の $m \in [M]$ に対して，

$$\|\boldsymbol{x}^\star_{\delta_m} - \boldsymbol{x}^\star\| < d_m|\delta_m|$$

が成り立つ．

(ii) 任意の $m \in [M]$ に対して $\gamma_m \in \left(\frac{1}{d_{m+1}}, 1\right)$ が成り立つならば，$m \in \{2, 3, \ldots, M\}$ に対して

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

が成り立つ．

命題 3 (i) は，関数 f が new σ-nice 関数であれば，常に関数 $\hat{f}_{\delta_m}$ の最適解がその関数の強凸領域に含まれることを意味している．命題 3 (ii) は，$\gamma_m \in \left(\frac{1}{d_{m+1}}, 1\right)$ が成り立つとき，関数 $\hat{f}_{\delta_m}$ の最適化の初期点が，その関数の強凸領域に含まれることを意味している．したがって，$\gamma_m \in \left(\frac{1}{d_{m+1}}, 1\right)$ が成り立てば，関数 $\hat{f}_{\delta_m}$ の最適化の初期点と最適解はどちらもその関数の強凸領域に含まれる．よって，最急降下法の点列が強凸領域を出ることはあり得ない．以上のことから，段階的最適化の初期点が，最初に最適化される関数 $\hat{f}_{\delta_1}$ の強凸領域 $N(\boldsymbol{x}^\star; d_1|\delta_1|)$ に含まれてさえいれば，段階的最適化アルゴリズムが目的関数 f の大域的最適解 $\boldsymbol{x}^\star$ に到達できるといえる．

式 (6) から，$1/d_m$ がとりうる値の範囲は

$$\frac{\sqrt{a_m^2-1}-1}{a_m} \leq \frac{1}{d_m} \leq 1 \tag{8}$$

であるとわかるから，γ_m がとりうる値の範囲は，図 2 の塗りつぶされた領域である．したがって，アルゴリズムによって生成される点列が強凸領域の外に出ないようにするためには，減衰率は非常にゆるかやに減少させる必要がある．また，ノイズが大きい，すなわち a_m が大きい学習初期は，ほとんど 1 と変わらないほど大きい減衰率 γ_m を使用するべきであることがわかる．よって，σ-nice 関数の定義においては，ノイズ

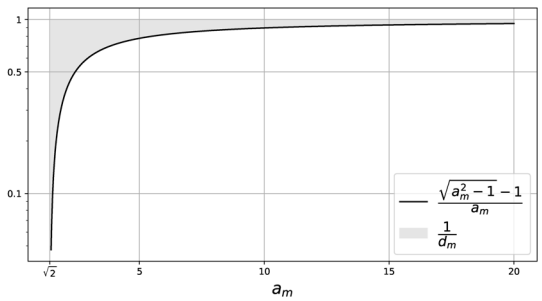

図 2　γ_m がとりうる値の範囲

の減衰率は m によらず $\gamma_m = 0.5$ であったが，これはノイズ $|\delta_m|$ が大きいときには成り立たないことがわかる．

9. 最適な学習率スケジューリング

命題 3 から，段階的最適化アルゴリズムにとって最適なノイズの減衰率 γ_m がわかった．また，式 (5) から，SGD の確率的ノイズによる平滑化の度合いは，学習率に比例することがわかっている．したがって，最適なノイズの減衰率はただちに最適な学習率の減衰率となる．図 3 は，既存の減少学習率スケジューラーの減衰率と，最適な減衰率の曲線を平行移動したものをプロットしたものである．

これによると，学習率の減衰率が満たすべき範囲に収まっているのは，次の式で定義される多項式減衰学習率のみである．

$$\eta_t := (\eta_{\max} - \eta_{\min})\left(1 - \frac{t}{T}\right)^p + \eta_{\min} \ (p > 0).$$

ただし，図 3 にプロットされた曲線は，$t \in [T], T = 200, \eta_{\min} = 0, \eta_{\max} = 0.1, p = 0.5$ と設定したものである．その他の減少学習率スケジューラーの定義については文献 [2] の 3.2 節を参照されたい．図 4 は，多項式減衰学習率のハイパーパラメータ p を変化させたときの減衰率と，最適な減衰率の範囲をプロットしたものである．これによると，1 以下の p を有する多項式減衰学習率が最適な減少学習率であるといえる．多項式減衰学習率の定義から，その減衰率は次のように計算することができる．

$$\gamma_m = \frac{(M-m)^p}{\{M-(m-1)\}^p}. \tag{9}$$

ただし，$m \in [M], p \in (0, 1]$ とする．

10. 暗黙的な段階的最適化アルゴリズム

3 節で示したとおり，訓練データの総数 n とモデルのパラメータの数 d が非常に大きいとき，式 (1) で定義される目的関数 f に対しては定義 1 の平滑化演算を施すことができないため，一般に段階的最適化アルゴリズムを DNN の訓練に適用することはできない．しかし，5 節で示したように，目的関数 f の最適化に SGD を使うというだけで，関数 f はある程度平滑化されていて，その度合い δ はハイパーパラメータである学習率 η とバッチサイズ b によって定まる．したがって，訓練中に平滑化の度合い δ が徐々に減少するように，学習率 η とバッチサイズ b を適切に変化させれば，

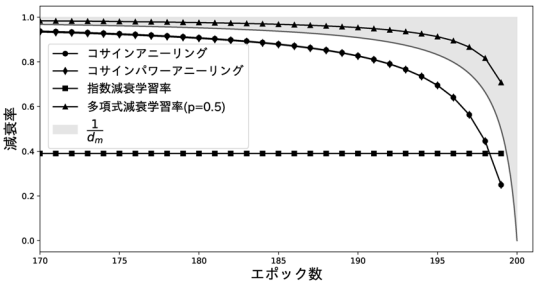

図 3　既存の学習率スケジューラーの減衰率

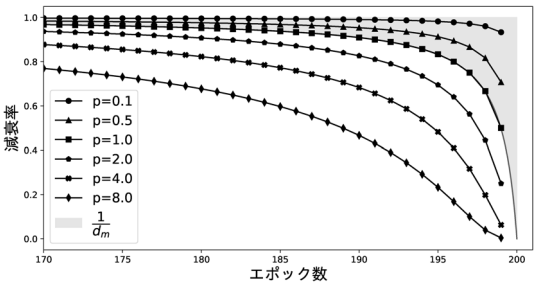

図 4　多項式減衰学習率の減衰率

SGD を利用した暗黙的な段階的最適化が実現できるはずである．さらに 8 節と 9 節から，減衰率 γ_m は式 (9) で設定すればよいことがわかっている．これらによって構成される暗黙的な段階的最適化アルゴリズムを次に示す．

Require: $\epsilon > 0, p \in (0, 1], \boldsymbol{x}_1 \in \mathbb{R}^d, \bar{d}, \eta_1, b_1 > 0, H_3, H_4 > 0$

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

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

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

次の定理によって，暗黙的な段階的最適化アルゴリズムの収束が保証される．証明は文献 [2] の D.4 節を参照されたい．

定理 1. $\epsilon > 0$ を十分小さい正数とする．このとき，L_f-リプシッツ new σ-nice 関数 f に対して暗黙的な段階的最適化アルゴリズムを適用すると，アルゴリズムは $\mathcal{O}\left(1/\epsilon^{\frac{1}{p}}\right)$ 回の反復で大域的最適解 $\boldsymbol{x}^\star$ の ϵ-近傍に到達する．

定理 1 は，非凸の目的関数 f が new σ-nice 関数ならば，暗黙的な段階的最適化アルゴリズムによって大域的最適解を見つけることができることを意味している．このアルゴリズムを画像分類タスクに適用した数値実験の結果は，文献 [2] の 4 章を参照されたい．

11. まとめ

本稿では，深層ニューラルネットワークを含む非凸関数の大域的最適化について，SGD を利用した段階的最適化の観点から考察した．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の条件を満たす非凸関数族である new σ-nice 関数を定義し，適切な大きさのノイズで平滑化すると，任意の関数が new σ-nice 関数となることを示した．また，SGD がもつ確率的ノイズの大きさは学習率，バッチサイズ，確率的勾配の分散によって定まることを示し，この性質を利用して，最適な学習率スケジューリングが多項式減衰学習率であることを示した．最後に，SGD の平滑化効果を利用した暗黙的な段階的最適化アルゴリズムと，その収束解析を紹介した．このアルゴリズムによって，SGD を利用した非凸関数の大域的最適化が可能となることを示すことができた．

謝辞 本稿を発表する機会を与えていただいた，筑波大学の吉瀬章子先生，学生優秀発表賞審査委員会の先生方，成蹊大学の関谷和之先生，法政大学の鮭川矩義先生に深く感謝申し上げます．なお，本研究は，日本学術振興会 科学研究費補助金 基盤研究 (C) (21K11773) の補助を受けています．

参考文献

[1] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, 1987.
[2] N. Sato and H. Iiduka, "Using stochastic gradient descent to smooth nonconcex functions: Analysis of implicit graduated optimization with optimal noise scheduling," *arXiv*, https://arxiv.org/abs/2311.08745, 2023.
[3] M. Nikolova, Michael K. Ng and Chi-Pan Tam, "Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction," *IEEE Transactions on Image Processing*, **19**, 2010.
[4] L. Peng, C. Kümmerle and R. Vidal, "On the cinvergence of IRLS and its variants in outlier-robust estimation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17808–17818, 2023.
[5] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pp. 11895–11907, 2019.
[6] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," In *Proceedings of the 32nd International Conference on Machine Learning*, **37**, pp. 2256–2265, 2015.
[7] J. Ho, A. Jain and P. Abbeel, "Denoising diffusion probabilistic models," In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pp. 6840–6851, 2020.
[8] H. Mobahi and J. W. Fisher III, "On the link between gaussian homotopy continuation and convex envelopes," In *Proceedings of the 10th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, **8932**, pp. 43–56, 2015.
[9] E. Hazan, K. Yehuda and S. Shalev-Shwartz, "On graduated optimization for stochastic non-convex problems," In *Proceedings of the 33rd International Conference on Machine Learning*, **48**, pp. 1833–1841, 2016.
[10] R. Johnson and Tong Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pp. 315–323, 2013.
[11] R. Ge, F. Huang, C. Jin and Y. Yuan, "Escaping from saddle points - online stochstic gradient for tensor decomposition," In *Proceedings of the Conference on Learning Theory*, **40**, pp. 797–842, 2015.
[12] 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 *Proceedings of the 5th International Conference on Learning Representations*, 2017.
[13] R. Kleinberg, Y. Li and Y. Yuan, "An alternative view: When does SGD escape local minima?" In *Proceedings of the 35th International Conference on Machine Learning*, **80**, pp. 2703–2712, 2018.
[14] J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. Kumar and S. Sra, "Why are adaptive methods good for attention models?" In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pp. 15383–15393, 2020.
[15] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, Cambridge University Press, 2018.
[16] E. Hoffer, I. Hubara and D. Soudry, "Train longer, generalize better: Closing the generalization gap in large batch training of neural networks," In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pp. 1731–1741, 2017.
[17] Y. You, J. Li, S. J. Reddi, J. Hseu, S. Kumar, S. Bhojanapali, X. Song, J. Demmel, K. Keutzer and C. Hsieh, "Large batch optimization for deep learning: Training BERT in 76 minutes," In *Proceedings of the*

8th International Conference on Learning Representations, 2020.

[18] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Learning*, **40**, pp. 834–848, 2018.

[19] S. L. Smith, P. Kindermans, C. Ying and Q. V. Le, "Don't decay the learning rate, increase the batch size," In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[20] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, **9**, pp. 1–42, 1997.

[21] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," In *Proceedings of the 34th Conference on Uncertainly in Artificial Intelligence*, pp. 876–885, 2018.

[22] H. Li, Z. Xu, G. Taylor, C. Studer and T. Goldstein, "Visualizing the loss landscape of neural nets," In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pp. 6391–6401, 2018.

