一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

SGDの確率的ノイズを利用した段階的最適化手法による経験損失関数の大域的最適化

佐藤尚樹†　　飯塚秀明†

† 明治大学
E-mail: †naoki310303@gmail.com, ††iiduka@cs.meiji.ac.jp

あらまし　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確率的勾配降下法 (SGD) について，段階的最適化の観点から考察する．段階的最適化とは，確率的ノイズによる平滑化演算を利用して非凸関数の大域的最適解を探索する手法である．本稿は SGD の確率的ノイズ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，確率的勾配の分散によって定まることを示す．この発見から，訓練中に学習率とバッチサイズを変化させることで暗黙的に平滑化の度合いを減少させる新しい段階的最適化アルゴリズムを提案し解析する．さらに，SGD の確率的ノイズによる平滑化の度合いと，深層学習モデルの汎化性能の間には密接な関係があることを実験的に示す．
キーワード　深層学習理論，非凸最適化，確率的勾配降下法，段階的最適化，平滑化

Naoki SATO† and Hideaki IIDUKA†

† Meiji University
E-mail: †naoki310303@gmail.com, ††iiduka@cs.meiji.ac.jp

Abstract　This paper discusses stochastic gradient descent (SGD) for solving empirical loss minimization problems that appear in deep learning from the viewpoint of graduated optimization. The graduated optimization is a method for finding global optimal solutions for nonconvex functions by using a function smoothing operation with stochastic noise. We show that stochastic noise in SGD has the effect of smoothing the objective function, the degree of which is determined by the learning rate, batch size, and variance of the stochastic gradient. Using this finding, we propose and analyze a new graduated optimization algorithm that varies the degree of smoothing by varying the learning rate and batch size. We further show that there is a close relationship between the degree of smoothing due to stochastic noise in SGD and the generalization performance of deep learning models.
Key words　deep learning theory, nonconvex optimization, stochastic gradient descent, graduated optimization, smoothing

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

$$\text{目的関数：}\ f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) \to \text{最小} \qquad (1)$$

$$\text{条件：}\ \boldsymbol{x} \in \mathbb{R}^d$$

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持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，

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

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，

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

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，

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

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 $\mathcal{S}_t \subset \mathcal{S}$ で次のように推定される．

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

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．

3. 段階的最適化

段階的最適化（Graduated optimization）[1] 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．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，図 2 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等の裾の軽い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．

［定義 1］(関数の平滑化) ある関数 f を平滑化して得られる関数 $\hat{f}_\delta\colon \mathbb{R}^d \to \mathbb{R}$ は，

$$\hat{f}_\delta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim\mathcal{L}}\left[f(\boldsymbol{x} - \delta\boldsymbol{u})\right]$$

と表される．ここで，$\delta \in \mathbb{R}$ は平滑化の度合いを表し，$\boldsymbol{u}$ は任意の裾の軽い分布 $\mathcal{L}$ に従う確率変数であり，$\mathbb{E}_{\boldsymbol{u}\sim\mathcal{L}}[\|\boldsymbol{u}\|] \leq 1$ を満たす．また，

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

とする．

段階的最適化手法はノイズ除去 [2] やロバスト推定 [3] 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 [4] と拡散モデル [5],[6] 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 [7] を参照されたい．

Algorithm 1 段階的最適化アルゴリズム

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

Hazan らは，上記のような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 [8].

［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 2 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 $\boldsymbol{x}^\star_\delta$ に対して，

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

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

(ii) 任意の $\delta > 0$ に対して，関数 $\hat{f}_\delta(\boldsymbol{x})$ は中心 $\boldsymbol{x}^\star_\delta$ で半径 3δ

の開近傍 $N(\boldsymbol{x}^\star_\delta; 3\delta)$ で σ-強凸である．

Hazan らは σ-nice 関数に対して段階的最適化アルゴリズムを使用すると，$O\left(1/\epsilon^2\right)$ 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 [8]．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

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

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ことが知られている [9]．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ことが経験的に知られている [15]．さらに，Kleignberg 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 [12]．

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，

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

とすると，

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

が成り立つ．式 (2) の導出は，文献 [14] の A 章を参照されたい．新たに関数 $\hat{f}(\boldsymbol{y}) := \mathbb{E}_{\omega_t}\left[f(\boldsymbol{y} - \eta\boldsymbol{\omega}_t)\right]$ を定義すれば，

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

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，

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

が成り立つ．証明は，文献 [14] 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，$\mathbb{E}_{\xi_t}\left[\|\boldsymbol{u}_t\|\right] \leq 1$ を満たす確率変数 $\boldsymbol{u}_t$ を使って，

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

と表すことができる．ここで，$\boldsymbol{\omega}_t$ が正規分布のような分布に従うという実験的観察から，$\boldsymbol{\omega}_t$ はある裾の軽い分布 $\hat{\mathcal{L}}$ に従うと仮定する．したがって，$\boldsymbol{\omega}_t \sim \hat{\mathcal{L}}$ と $\boldsymbol{u}_t \sim \mathcal{L}$ が成り立つ．ただし，$\hat{\mathcal{L}}$ と $\mathcal{L}$ は裾の軽い分布で，$\mathcal{L}$ は $\hat{\mathcal{L}}$ をスケールした分布である．確率的ノイズが正規分布のような分布に従うことについては，文献 [13] や文献 [14] の H 章を参照されたい．これを利用して式 (2) をさらに変形すると，

$$\begin{aligned}\mathbb{E}_{\omega_t}\left[\boldsymbol{y}_{t+1}\right] &= \mathbb{E}_{\omega_t}\left[\boldsymbol{y}_t\right] - \eta\nabla\mathbb{E}_{\omega_t}\left[f(\boldsymbol{y}_t - \eta\boldsymbol{\omega}_t)\right]\\ &= \mathbb{E}_{\omega_t}\left[\boldsymbol{y}_t\right] - \eta\nabla\mathbb{E}_{\boldsymbol{u}_t\sim\mathcal{L}}\left[f\left(\boldsymbol{y}_t - \frac{\eta C}{\sqrt{b}}\boldsymbol{u}_t\right)\right]\\ &= \mathbb{E}_{\omega_t}\left[\boldsymbol{y}_t\right] - \eta\nabla\hat{f}_{\frac{\eta C}{\sqrt{b}}}(\boldsymbol{y}_t).\end{aligned} \tag{4}$$

が成り立つ．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，

$$\delta = \frac{\eta C}{\sqrt{b}} \tag{5}$$

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 [15] 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．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 [18], [20] 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．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 [15], [19], [21], [22] に基づいていることに注意したい（図 1 を参照）．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，厳密な理論的証明はない．

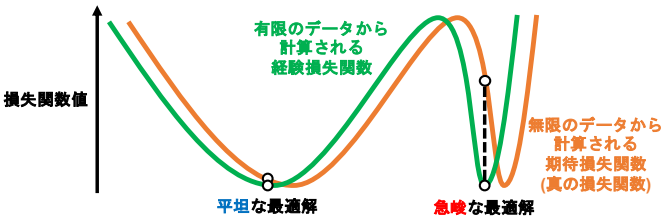

図 1　経験損失関数の形状の概念図

7. σ_m-nice 関数

次節では，第 5 節で明らかになった SGD の平滑化特性を利用した新たな段階的最適化アルゴリズムを提案する．その準備として，Hazan らによって提案された σ-nice 関数をやや拡張した σ_m-nice 関数を導入する．

［定義 3］(σ_m-nice function)　$M \in \mathbb{N}, m \in [M], \gamma \in [0.5, 1)$ とする．任意の関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ に対して，次の 2 つの条件が満たされるとき，関数 f は σ_m-nice 関数であるという．

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

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

を満たすような $\boldsymbol{x}^\star_{\delta_{m+1}}$ が存在する．

(ii) 任意の $\delta_m > 0$ に対して，関数 $\hat{f}_{\delta_m}(\boldsymbol{x})$ は開近傍 $N(\boldsymbol{x}^\star_{\delta_m}; 3\delta_m)$ で σ_m-強凸である．

σ_m-nice 関数がどのように σ-nice 関数を拡張しているかを説明しよう．まず条件 (i) において，平滑化の度合いの減衰率は，σ-nice 関数では常に 0.5 であったところを，σ_m-nice 関数ではやや一般化された $\gamma \in [0.5, 1)$ となっている．この拡張は段階的最適化アルゴリズムの収束保証を損なわない．詳細は文献 [14] の Proposition 5.1 を参照されたい．次に，条件 (ii) において，平滑化された関数 $\hat{f}_{\delta_m}$ は，σ-nice 関数では平滑化の度合い δ_m によらず常に σ-強凸であったところを，σ_m-nice 関数では m に依存する σ_m-強凸となっている．強凸関数の定数は，関数がより滑らかな方が小さくなるため，例えば $\hat{f}_{\delta_1}$ と $\hat{f}_{\delta_m}$ では異なる定数を設定するべきである．実際，$0 < \sigma_{\text{small}} < \sigma_{\text{big}}$ とし，ある関数 g が σ_{small}-強凸であるとき，関数 g は σ_{big}-強凸ではない．したがって任意の δ_m に対して $\hat{f}_{\delta_m}$ が σ-強凸である σ-nice 関数の定義は現実的ではない．σ_m-nice 関数の定義は，この問題を回避している．

8. 暗黙的な段階的最適化アルゴリズム

3 節で示した通り，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，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．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 (その概念図については，図 2 を参照)．

Algorithm 2 暗黙的な段階的最適化アルゴリズム

Require: $\epsilon, \boldsymbol{x}_1 \in B(\boldsymbol{x}^\star_{\delta_1}; 3\delta_1), \eta_1 > 0, b_1 \in [n], \gamma \geq 0.5$

$\delta_1 := \frac{\eta_1 C}{\sqrt{b_1}}$, $\alpha_0 := \min\left\{\frac{1}{16 L_f \delta_1}, \frac{1}{\sqrt{2}\sigma\delta_1}\right\}$, $M := \log_\gamma \alpha_0\epsilon$
for $m = 1$ to $M + 1$ **do**
  if $m \neq M + 1$ **then**
    $\epsilon_m := \sigma_m \delta_m^2/2$, $T_m := H_m/\epsilon_m$
    $\kappa_m/\sqrt{\lambda_m} = \gamma$ $(\kappa_m \in (0, 1], \lambda_m \geq 1)$
  end if
  $\boldsymbol{x}_{m+1} := \mathrm{GD}(T_m, \boldsymbol{x}_m, \hat{f}_{\delta_m}, \eta_m)$
  $\eta_{m+1} := \kappa_m \eta_m, b_{m+1} := \lambda_m b_m$
  $\delta_{m+1} := \frac{\eta_{m+1} C}{\sqrt{b_{m+1}}}$
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 [14] 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 σ_m-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 $O\left(1/\epsilon^2\right)$ 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 σ_m-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．最後に，暗黙的な段階的最適化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を紹介しよう．図 3 は，ImageNet データセットによる ResNet34 の 200 エポックの訓練における，テスト精度と訓練損失関数値をプロットしたものである．手法 1 は定数学習率および定数バッチサイズを有する通常の SGD であり，手法 2 は学習率のみを減衰させる SGD, 手法 3 はバッチサイズのみを増加させる SGD, 手法 4 は学習率を減衰させ，バッチサイズは増加させる SGD である．すなわち手法 2,3,4 は暗黙的な段階的最適化アルゴリズムである．学習率とバッチサイズは，平滑化の度合いが 40 エポックごとに $\frac{1}{\sqrt{2}}$ 倍になるように更新さ

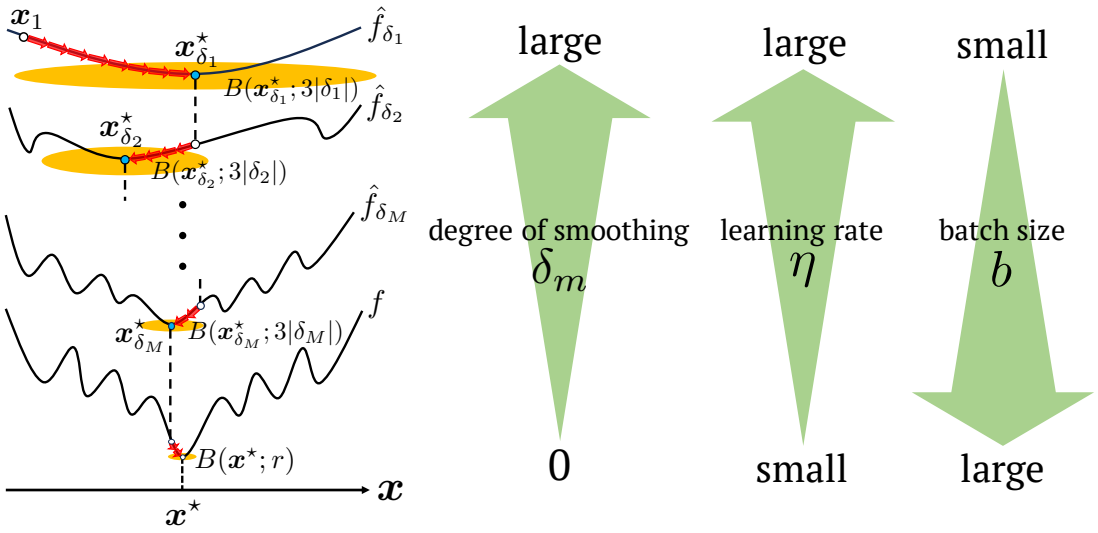

図 2　σ_m-nice 関数に対する暗黙的な段階的最適化アルゴリズムの概念図

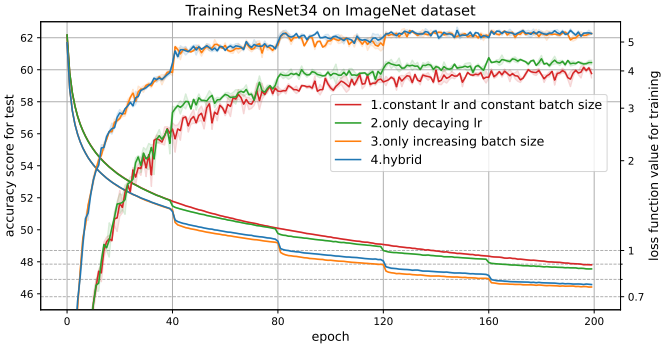

図 3　ImageNet データセットによる ResNet34 の訓練におけるテスト精度と訓練損失関数値．実線は 3 回の実行の平均値を表しており，3 回の実行の最大値と最小値の間を薄く塗りつぶしている．

れた．すなわち，$\gamma = \frac{1}{\sqrt{2}}$ である．より詳細な設定については，文献 [14] の 5 章を参照されたい．図 3 は暗黙的な段階的最適化アルゴリズムが通常の SGD よりも高いテスト精度と低い訓練損失関数値を達成できていることを表している．

9. 平滑化の度合いと Sharpness と汎化性能の関係

前節までは，訓練中に平滑化の度合いを減少させ，暗黙的な段階的最適化を実現するための考察をしてきた．本節では，訓練中に平滑化の度合いが変化しない，すなわち，定数学習率，定数バッチサイズの場合を考察する．

6 節で述べた通り，最適化アルゴリズムが収束する近似解の近傍での経験損失関数の形状は，深層学習モデルの汎化性能と密接な関係があると考えられてきた（図 1 も参照）．その関係を探るために，いくつかの先行研究 [15],[23]〜[26] は近似解の周りの関数の鋭さの指標として “Sharpness” を提案している．特に，Kwon らが提案した “adaptive sharpness” は，訓練データの集合 $\mathcal{S}$，任意のモデルのパラメータ $\boldsymbol{w} \in \mathbb{R}^d$ とあるベクトル $\boldsymbol{c} \in \mathbb{R}^d$ に対して，

$$S^\rho_{\max}(\boldsymbol{w}, \boldsymbol{c}) := \mathbb{E}_{\mathcal{S}}\left[\max_{\|\boldsymbol{\delta} \odot \boldsymbol{c}^{-1}\|_p \le \rho} f(\boldsymbol{w}+\boldsymbol{\delta}) - f(\boldsymbol{w})\right]$$

で定義される．ただし，$\rho \in \mathbb{R}$ は近傍の半径を表しており，$\odot/^{-1}$ はそれぞれ要素ごとの積と商の計算を表している．したがって，Sharpness の値が大きければ大きいほど，パラメータ $\boldsymbol{w}$ の周りでの関数の形状は鋭いことを意味する．経験損失関数の滑らかさを議論するときに，Sharpness を欠かすことはできない．5 節では，SGD の確率的ノイズによる平滑化の度合いが $\delta = \frac{\eta C}{\sqrt{b}}$ で表せることを示した．それでは，平滑化の度合い δ と Sharpness の間にはどのような関係があるだろうか．本稿はこれを確かめるために，ResNet18 を CIFAR100 データセットで 200 エポック，学習率 $\eta \in \{0.01, 0.05, 0.1, 0.5\}$ とバッチサイズ $b \in \{2^1, \ldots, 2^{13}\}$ の組み合わせで訓練し，テスト精度を計測する．また，得られた近似解の周りの adaptive sharpness を，$\rho = 0.0002$ と $\boldsymbol{c} := (1, 1, \ldots, 1)^\top \in \mathbb{R}^d$ に対して計測した．これらの測定は学習率とバッチサイズの組み合わせ 1 つに対して 3 回行われ，計 156 個のテスト精度と Sharpness の値が集まった．一方，平滑化の度合い $\delta = \frac{\eta C}{\sqrt{b}}$ はその定義に従って，訓練に使われた学習率 η とバッチサイズ b を使って計算した．ここで，確率的勾配の分散 C^2 は推定値を利用している．推定の詳細については文献 [14] の B 章を参照されたい．図 4 は，全ての実験結果をまとめたものである．なお，図中の “lr” は学習率を意味している．

図 4(a) は，バッチサイズが大きいほど Sharpness が大きくなること，(b) は学習率が大きいほど Sharpness が小さくなること，(c) は Sharpness が小さいほど平滑化の度合いが大きくなることを示している．これらの結果は，平滑化の度合い δ が学習率 η に比例し，バッチサイズ b に反比例するという我々の理論結果を保証し，$\delta = \frac{\eta C}{\sqrt{b}}$ が関数の平滑化の度合いを決定しているという我々の理論を補強している．また，Keskar ら [15] は，大きすぎるバッチサイズが Shapness の増加をもたらし，汎化性能を悪化させることを実験的に観察している．図 4(a) はこ

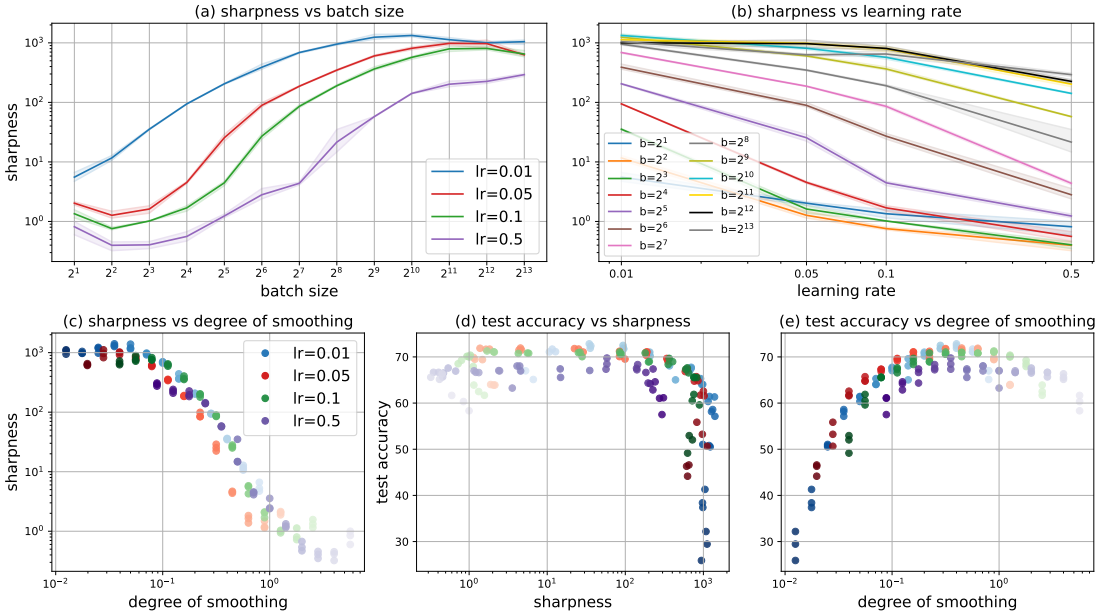

図 4　**(a)** CIFAR100 データセットによる ResNet18 の 200 エポックの訓練で得られた近似解の近傍で計測された Sharpness と，訓練に使われたバッチサイズの関係．**(b)** Sharpness と，訓練に使われた学習率の関係．**(c)** Sharpness と，学習率，バッチサイズ，推定された確率的勾配の分散から計算された平滑化の度合いの関係．**(d)** テスト精度と Sharpness の関係．**(e)** テスト精度と平滑化の度合いの関係．実線は 3 回の実行の平均値を表しており，3 回の実行の最大値と最小値の間を薄く塗りつぶしている．散布図における色の濃淡はバッチサイズの大小を表しており，色が濃いほどバッチサイズが大きいことを意味する．全てのグラフは同じ実験結果を異なる視点から示している．

の実験的観察と一致し，この現象も我々の理論によって理論的な説明を与えることができる．すなわち，大きすぎるバッチサイズが Sharpness の増加をもたらすのは，バッチサイズが平滑化の度合いに対して反比例するためであるといえる．図 4(d) は，モデルの汎化性能と近似解付近の Sharpness との間に明確な相関がないことを示している．この結果は，ResNet 等の現代的な深層学習モデルの訓練においては，Sharpness と汎化性能の間には相関が見られないことを示した先行研究 [27] と一致する．一方，図 4(e) は，汎化性能と平滑化の度合いとの間に明確な関係がある（汎化性能は平滑化の度合いに対して凹関数になっている）ことを示しており，大きすぎず小さすぎない平滑化の度合いが高い汎化性能をもたらすことを示している．

Andriushchenko ら [27] は，Sharpness と汎化性能に関する既存の先行研究がいずれも小規模な深層学習モデルを対象にしていることに注目し，現代的な大規模な深層学習モデルを用いた広範な数値実験を行い，それらに対しては Sharpness と汎化性能の間にはほぼ相関が見られないことを発見し，Sharpness は汎化性能の良い指標ではない可能性を示唆した．またそれによって，「より平坦な最適解がより良い汎化性能をもたらす」という仮説の正しさに疑問符を投げかけている．図 4(c) から，平滑化の度合いと Sharpness の両方が目的関数の滑らかさ/鋭さを表していることは明らかである．さらに，図 4(d) と (e) から，上述の先行研究と同様に，Sharpness と汎化性能の間には相関がない一方で，平滑化の度合いと汎化性能の間には相関があることが分かる．したがって，平滑化の度合いは汎化性能を理論的に評価するためのより良い指標であるといえ，またそのため，「より平坦な最適解がより良い汎化性能をもたらす」という仮説を破棄するのは早計かもしれない．

10. 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．SGD が持つ確率的ノイズには目的関数を平滑化する効果があることを示し，その平滑化の度合いは学習率，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た．さらに，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した．最後に，SGD の確率的ノイズによる平滑化の度合いと，Sharpness とモデルの汎化性能の間には密接な関係があることを実験的に示した．この結果は，「より平坦な最適解がより良い汎化性能をもたらす」という主張を支持するものである．

文　献

[1] A. Blake and A. Zisserman,“Visual Reconstruction,” *MIT Press*, 1987.
[2] M. Nikolova, Michael K. Ng, and Chi-Pan Tam, “Fast nonconvex non-smooth minimization methods for image restoration and reconstruction,” *IEEE Transactions on Image Processing*, **19**, 2010.
[3] 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.
[4] 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.

[5] 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.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, 2020.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[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] 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.

[13] 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*, 2020.

[14] N. Sato and H. Iiiduka, "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.

[15] 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 minim," In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[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. Hochreiter and J. Schmidhuber, "Flat minima", *MIT Press*, **9**, pp. 1-42, 1997.

[20] 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.

[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.

[23] T. Liang, T. A. Poggio, A. Rakhlin, and J. Stokes, "Fisher-rao metric, geometry, and complexity of neural networks," In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, **89**, pp. 888-896, 2019.

[24] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis," In *Proceedings of the 37th International Conference on Machine Learning*, **119**, pp. 9636–9647, 2020.

[25] H. Petzka, M. Kamp, L. Adilova, C. Sminchisescu, and M. Boley, "Relative flatness and generalization," In *Proceedings of the 34th Conference on Neural Information Processing Systems*, pp. 18420-18432, 2021.

[26] J. Kwon, J. Kim, H. Park, and I. K. Choi, "ASAM: adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks," In *Proceedings of the 38th International Conference on Machine Learning*, **139**, pp.5905-5014, 2021.

[27] M. Andriushchenko, F. Croce, M. Müller, M. Hein, and N. Flammarion, "A modern look at the relationship between sharpness and generalization", In *Proceedings of the 40th International Conference on Machine Learning*, **202**, pp. 840-902, 2023.