適応勾配法を利用したニューラルネットワークの訓練

明治大学大学院理工学研究情報科学専攻　朱　伊妮
明治大学大学院理工学研究科情報科学専攻　酒井　裕行
明治大学理工学部情報科学科　飯塚　秀明
Yini Zhu, Hiroyuki Sakai
Computer Science Course,
Graduate School of Science and Technology,
Meiji University
Hideaki Iiduka
Department of Computer Science,
School of Science and Technology,
Meiji University

Abstract

本論文は、深層ニューラルネットワークを訓練するために必要な非凸確率的最適化に対する手法を提案する。提案アルゴリズムは既存の適応学習率最適化アルゴリズムとして知られている Adam、AMSGrad、GWDC、AMSGWDC を例としてもつように構成されている。本論文では、定数及び減少学習率に対する提案アルゴリズムの収束解析を与える。定数学習率を利用する場合、提案アルゴリズムは非凸確率的最適化問題の停留点をを近似することができる。減少学習率を利用する場合、提案アルゴリズムは問題の停留点に収束する。提案収束解析により、既存適応学習率最適化アルゴリズムは深層ニューラルネットワークに現れる非凸確率的最適化に適用可能であることが保証される。

1　はじめに

深層ニューラルネットワークに現れる非凸確率的最適化問題を解くことで深層ニューラルネットワークのパラメータを適切に調整することができる [3, 6, 8]。この問題を解くためのアルゴリズムは多く提案されている。特に、適応学習率最適化アルゴリズム [2, Subchapter 8.5] は高速に問題を解くことが可能な手法として知られている。例えば、AdaGrad [1]、RMSProp [2, Algorithm 8.5]、Adam [5]、AMSGrad [7]、GWDC [6, Algorithm 2]、AMSGWDC [6, Algorithm 3]、といった手法である。これらの手法は、共通してある種の正定値対称行列の逆行列を利用している。

本論文では、その正定値対称行列の性質（仮定 3.1）を紹介し、その性質のもとで収束が保証されるアルゴリズム [4]（Algorithm 1）を提案する。提案アルゴリズムは、Adam や AMSGrad といった既存手法の統一形となっている（例 3.1 を参照せよ)。定数学習率を有するアルゴリズムは非凸確率的最適化問題の停留点を近似することが可能である（定理 3.1）のに対して、減少学習率を有するアルゴリズムは問題の停留点に収束する（定理 3.2)。

2　深層ニューラルネットワークに現れる最適化

2.1　数学的準備

$\mathbb{N}$ を 0 とすべての正整数の集合とし、$\mathbb{R}^d$ を d 次元ユークリッド空間とする。$\mathbb{R}^d$ の内積は $\langle\cdot,\cdot\rangle$ とし、ノルムは $\|\cdot\|$ とする。$\mathbb{R}^d_{++} := \{x = (x_i) \in \mathbb{R}^d \colon x_i > 0\ (i = 1, 2, \ldots, d)\}$ と定義する。$\mathbb{S}^d$ は、$d \times d$ 対称行列全体の集合、すなわち、$\mathbb{S}^d = \{X \in \mathbb{R}^{d\times d} \colon X = X^\top\}$ である。$\mathbb{S}^d_{++}$ は、$d \times d$ 正定値対称行列全体の集合 $\mathbb{S}^d_{++} = \{X \in \mathbb{S}^d \colon X \succ O\}$ とする。$\mathbb{D}^d$ は、$d \times d$ 対角行列とし, $\mathbb{D}^d = \{X \in \mathbb{R}^{d\times d} \colon X = \mathsf{diag}(x_i),\ x_i \in \mathbb{R}\ (i = 1, 2, \ldots, d)\}$ とする。$A \odot B$ は、行列 A と B の Hadamard 積とする。このとき、任意の $x := (x_i) \in \mathbb{R}^d$ に対して、$x \odot x := (x_i^2) \in \mathbb{R}^d$ である。与えられた $H \in \mathbb{S}^d_{++}$ に対して、$\mathbb{R}^d$ 上の H-内積と H-ノルムは、任意の $x, y \in \mathbb{R}^d$ に対して、$\langle x, y\rangle_H := \langle x, Hy\rangle$ と $\|x\|_H^2 := \langle x, Hx\rangle$ で定義される。

空でない閉凸集合 X $(\subset \mathbb{R}^d)$ 上の距離射影 P_X は、任意の $x \in \mathbb{R}^d$ に対して、$P_X(x) \in X$ 及び $\|x - P_X(x)\| = \inf_{y\in X} \|x - y\|$ で定義される。H-ノルムのもとでの X 上の距離射影は $P_{X,H}$ と書く。確率変数 X の期待値を $\mathbb{E}[X]$ とする。

2.2　深層ニューラルネットワークに現れる非凸最適化

深層ニューラルネットワークに現れる非凸最適化問題は以下のように与えられる。

問題 2.1 以下を仮定する。

(A1) $X \subset \mathbb{R}^d$ は空でない閉凸集合であり、その距離射影は計算可能であるとする。

(A2) $F(\cdot, \xi)\colon \mathbb{R}^d \to \mathbb{R}$ は $\xi \in \Xi$ に対して、連続微分可能であるとする。ただし、$\xi \in \Xi$ は確率分布 P のもとでの確率変数であるとする。関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ は、任意の $x \in \mathbb{R}^d$ に対して、$f(x) := \mathbb{E}[F(x, \xi)]$ で定義される。

このとき、f の X 上での最小解を見つけたい。すなわち、以下を満たす $x^\star \in X$ を見つけたい。

$$x^\star \in \operatorname*{argmin}_{x \in X} f(x).$$

本論文では、問題 2.1 の停留点問題 [4] を考察する。

問題 2.2 仮定 *(A1)*、*(A2)* のもとで、問題 *2.1* の停留点を見つけよ。すなわち、以下の $x^\star \in X$ を見つけよ。

$$x^\star \in X^\star := \left\{ x^\star \in X\colon \langle x - x^\star, \nabla f(x^\star) \rangle \geq 0 \ (x \in X) \right\}.$$

ただし、∇f は f の勾配を表す。

問題 2.1 と問題 2.2 の関係は、以下の通りである。

(F1) 一般に、$\operatorname{argmin}_{x \in X} f(x) \subset X^\star$ である。

(F2) もしも f が凸ならば、$\operatorname{argmin}_{x \in X} f(x) \supset X^\star$ である。すなわち、$\operatorname{argmin}_{x \in X} f(x) = X^\star$ が成り立つ。

問題 2.2 は以下の条件下で解析される。

(C1) 確率変数 ξ の独立同一分布なサンプル $\xi_0, \xi_1, \ldots$ が存在するとする。

(C2) 入力点 $(x, \xi) \in \mathbb{R}^d \times \Xi$ に対して、$\mathbb{E}[\mathsf{G}(x, \xi)] = \nabla f(x)$ を満たす確率勾配 $\mathsf{G}(x, \xi)$ が得られるとする。

(C3) 任意の $x \in X$ に対して、$\mathbb{E}[\|\mathsf{G}(x, \xi)\|^2] \leq M^2$ を満たす正定数 M が存在するとする。

3　適応学習率最適化アルゴリズム

以下が提案アルゴリズムである。

Algorithm 1 問題 2.2 に対する適応学習率最適化アルゴリズム [4]

Require: $(\alpha_n)_{n\in\mathbb{N}} \subset (0,1)$, $(\beta_n)_{n\in\mathbb{N}} \subset [0,1)$, $\gamma \in [0,1)$

$n \leftarrow 0$, $x_0, m_{-1} \in \mathbb{R}^d$, $\mathsf{H}_0 \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$

loop

$m_n := \beta_n m_{n-1} + (1-\beta_n)\mathsf{G}(x_n, \xi_n)$

$\hat{m}_n := \dfrac{m_n}{1-\gamma^{n+1}}$

$\mathsf{H}_n \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$

Find $\mathsf{d}_n \in \mathbb{R}^d$ that solves $\mathsf{H}_n \mathsf{d} = -\hat{m}_n$

$x_{n+1} := P_{X,\mathsf{H}_n}(x_n + \alpha_n \mathsf{d}_n)$

$n \leftarrow n+1$

end loop

アルゴリズム 1 を解析するための以下の仮定を必要とする。

仮定 3.1 $\mathsf{H}_n := \mathsf{diag}(h_{n,i})$ で定義される行列の数列 $(\mathsf{H}_n)_{n\in\mathbb{N}} \subset \mathbb{S}^d_{++} \cap \mathbb{D}^d$ は以下を満たすとする。

(A3) 任意の $n \in \mathbb{N}$, $i = 1, 2, \ldots, d$ に対して、$h_{n+1,i} \geq h_{n,i}$.

(A4) 任意の $i = 1, 2, \ldots, d$ に対して、正数 B_i が存在して、$\sup\{\mathbb{E}[h_{n,i}] \colon n \in \mathbb{N}\} \leq B_i$.

さらに、

(A5) $D := \max_{i=1,2,\ldots,d} \sup\{(x_i - y_i)^2 \colon (x_i), (y_i) \in X\} < +\infty$.

例 3.1

(i) Adam [5]: 以下で定義される H_n と v_n $(n \in \mathbb{N})$ を考察する。

$$
\begin{aligned}
&v_n := \delta v_{n-1} + (1-\delta)\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n), \\
&\overline{v}_n := \frac{v_n}{1-\delta^{n+1}}, \\
&\hat{v}_n = (\hat{v}_{n,i}) := \left(\max\{\hat{v}_{n-1,i}, \overline{v}_{n,i}\}\right), \\
&\mathsf{H}_n := \mathsf{diag}\left(\sqrt{\hat{v}_{n,i}}\right).
\end{aligned}
\tag{1}
$$

ただし、$v_{-1} = \hat{v}_{-1} = 0 \in \mathbb{R}^d$, $\delta \in [0, 1)$ であるとする。(1) で定義される H_n and v_n は (A3), (A4) を満たす [4, Section 3]。

(ii) AMSGrad [7]:

$$\begin{aligned} &v_n := \delta v_{n-1} + (1-\delta)\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n), \\ &\hat{v}_n = (\hat{v}_{n,i}) := (\max\{\hat{v}_{n-1,i}, v_{n,i}\}), \\ &\mathsf{H}_n := \mathsf{diag}\left(\sqrt{\hat{v}_{n,i}}\right) \end{aligned} \tag{2}$$

で定義される H_n と v_n $(n \in \mathbb{N})$ は (A3), (A4) を満たす [4, Section 3]。ただし、$v_{-1} = \hat{v}_{-1} = 0 \in \mathbb{R}^d$, $\delta \in [0, 1)$ である。(2) から成る Algorithm 1 は、AMSGrad アルゴリズム [7] である。

(iii) GWDC [6]: $(l_n)_{n\in\mathbb{N}} \subset \mathbb{R}_{++}$ は単調増加するとし、$(u_n)_{n\in\mathbb{N}} \subset \mathbb{R}_{++}$ は単調減少するとする。また、任意の $n \in \mathbb{N}$ に対して、$l_n \leq u_n$ を満たすとする。以下で定義される H_n と v_n $(n \in \mathbb{N})$ を考察する。

$$\begin{aligned} &v_n := \delta v_{n-1} + (1-\delta)\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n), \\ &\hat{v}_n = (\hat{v}_{n,i}) := \left(\mathrm{Clip}\left(\frac{1}{\sqrt{v_{n,i}}}, l_n, u_n\right)^{-1}\right), \\ &\mathsf{H}_n := \mathsf{diag}\left(\sqrt{\hat{v}_{n,i}}\right). \end{aligned} \tag{3}$$

ただし、$v_{-1} = 0 \in \mathbb{R}^d$, $\delta \in [0, 1)$ であり、$\mathrm{Clip}(\cdot, l, u) \colon \mathbb{R} \to \mathbb{R}$ (ただし、$l \leq u$ を満たす $l, u \in \mathbb{R}$) は任意の $x \in \mathbb{R}$ に対して、

$$\mathrm{Clip}(x, l, u) := \begin{cases} l & \text{if } x < l, \\ x & \text{if } l \leq x \leq u, \\ u & \text{if } x > u \end{cases}$$

で定義されるものとする。明らかに、(3) で定義される H_n と v_n は (A3) を満たす（[6, (13)] も見よ)。さらに、任意の $n \in \mathbb{N}$ に対して、$l_0 \leq l_n \leq \mathrm{Clip}(1/\sqrt{v_{n,i}}, l_n, u_n) \leq u_n \leq u_0$ である。すなわち、(A4) を満たす。(3) から成る Algorithm 1 は、GWDC アルゴリズム [6, Algorithm 2] である。

(iv) AMSGWDC [6]: $(l_n)_{n\in\mathbb{N}}$, $(u_n)_{n\in\mathbb{N}}$ は上記 (iii) にある条件を満たすとする。H_n, v_n $(n \in \mathbb{N})$ を以下で定義する。

$$
\begin{aligned}
&v_n := \delta v_{n-1} + (1-\delta)\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n),\\
&\hat{v}_n = (\hat{v}_{n,i}) := \left(\max\{\hat{v}_{n-1,i}, v_{n,i}\}\right),\\
&\tilde{v}_n = (\tilde{v}_{n,i}) := \left(\mathrm{Clip}\left(\frac{1}{\sqrt{\hat{v}_{n,i}}}, l_n, u_n\right)^{-1}\right),\\
&\mathsf{H}_n := \mathsf{diag}\left(\sqrt{\tilde{v}_{n,i}}\right).
\end{aligned}
\tag{4}
$$

ただし、$v_{-1} = \hat{v}_{-1} = 0 \in \mathbb{R}^d$, $\delta \in [0, 1)$ である。例 3.1(ii), (iii) から、(4) で定義される H_n, v_n は (A3), (A4) を満たす。(4) から成る Algorithm 1 は、AMSGWDC アルゴリズム [6, Algorithm 3] である。

3.1 定数学習率を有する Algorithm 1 の収束解析

以下の定理は、定数学習率を有する Algorithm 1 の収束解析である。証明については、[4] を参照せよ。

定理 3.1 *(A1)–(A5), (C1)–(C3)* が成り立つとし、$(x_n)_{n\in\mathbb{N}}$ は、$\alpha_n := \alpha$, $\beta_n := \beta$ *$(n \in \mathbb{N})$* を有する *Algorithm 1* で生成される点列とする。このとき、任意の $x \in X$ に対して、

$$
\limsup_{n\to+\infty} \mathbb{E}\left[\langle x - x_n, \nabla f(x_n)\rangle\right] \geq -\frac{\tilde{B}^2\tilde{M}^2}{2\tilde{b}\tilde{\gamma}^2}\alpha - \frac{\tilde{M}\sqrt{Dd}}{\tilde{b}\tilde{\gamma}}\beta
$$

を満たす。ただし、$\tilde{\gamma} := 1 - \gamma$, $\tilde{b} := 1 - \beta$, $\tilde{M}^2 := \max\{\|m_{-1}\|^2, M^2\}$, D は *(A5)* で定義されており、$\tilde{B} := \sup\{\max_{i=1,2,\ldots,d} h_{n,i}^{-1/2} : n \in \mathbb{N}\} < +\infty$ とする。

3.2 減少学習率を有する Algorithm 1 の収束解析

以下の定理は、減少学習率を有する Algorithm 1 の収束解析である。証明については、[4] を参照せよ。

定理 3.2 *(A1)–(A5), (C1)–(C3)* を満たすとし、$(\alpha_n)_{n\in\mathbb{N}}$ と $(\beta_n)_{n\in\mathbb{N}}$ は

$$
\sum_{n=0}^{+\infty} \alpha_n = +\infty, \quad \sum_{n=0}^{+\infty} \alpha_n^2 < +\infty, \quad \sum_{n=0}^{+\infty} \alpha_n\beta_n < +\infty
$$

を満たす点列とする。また、$(x_n)_{n\in\mathbb{N}}$ は *Algorithm 1* で生成される点列とする。このとき、任意の $x \in X$ に対して、

$$\limsup_{n\to+\infty} \mathbb{E}\left[\langle x - x_n, \nabla f(x_n)\rangle\right] \geq 0$$

を満たす。

定理 3.1 及び定理 3.2 から Algorithm 1 の問題 2.2 への適用が保証される。

References

[1] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[2] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, 2016.

[3] M. A. Hanif, A. Manglik, and M. Shafique. Resistive crossbar-aware neural network design and optimization. *IEEE Access*, 8:229066–229085, 2020.

[4] H. Iiduka. Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. *arXiv:2002.09647*, 2020.

[5] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *Proceedings of The International Conference on Learning Representations*, pages 1–15, 2015.

[6] D. Liang, F. Ma, and W. Li. New gradient-weighted adaptive gradient methods with dynamic constraints. *IEEE Access*, 8:110929–110942, 2020.

[7] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. *Proceedings of The International Conference on Learning Representations*, pages 1–23, 2018.

[8] Y. Yu and F. Liu. Effective neural network training with a new weighting mechanism-based optimization algorithm. *IEEE Access*, 7:72403–72410, 2019.