深層ニューラルネットワークを訓練する際に利用する適応学習率最適化アルゴリズムの適切な勾配

明治大学大学院理工学研究科情報科学専攻　下山　歌奈子
明治大学理工学部情報科学科　飯塚　秀明
Kanako Shimoyama,
Computer Science Course, Graduate School of Science and Technology,
Meiji University
Hideaki Iiduka
Department of Computer Science, School of Science and Technology,
Meiji University

概要

本論文では、深層学習に現れる確率的非凸最適化問題を扱い、深層ニューラルネットワークを訓練するための適応学習率最適化アルゴリズムについて考察する。特に、適応学習率最適化アルゴリズム内で利用される勾配が、アルゴリズムの高速収束についてどのように影響するかについて考察する。

第 1 章　はじめに

深層学習の主な目的の一つとして、深層ニューラルネットワークを適切に学習する事が挙げられる。この目的を達成するための方法として、期待リスクや経験リスクと呼ばれる特定のコスト関数を減少させるような深層ニューラルネットワークモデルを見つける方法がある。そのとき、コスト関数を最小化にするための最適化手法が必要となろう。本論文では、適応学習率最適化アルゴリズムと呼ばれる深層ニューラルネットワークを訓練するための最適化手法について考察をする。

本論文では、適応学習率最適化アルゴリズム内で利用される勾配が、アルゴリズムの高速収束についてどのように影響するかについて考察する。

第 2 章　非凸最適化問題における停留点問題

2.1 数学的準備

$\mathbb{N}$ は全ての正数と 0 をから成る集合であり、$\mathbb{R}^d$ は内積を $\langle \cdot, \cdot \rangle$、ノルムを $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$ で定義する d 次元のユークリッド空間とする。$\mathbb{S}^d = \{M \in \mathbb{R}^{d \times d} \colon M = M^\top\}$ で定義され

る $d \times d$ 対称行列全体の集合とし、$\mathbb{S}^d_{++} = \{M \in \mathbb{S}^d \colon M \succ O\}$ で定義される $d \times d$ 正定値対称行列全体の集合とする。$\mathbb{D}^d = \{M \in \mathbb{R}^{d\times d} \colon M = \mathsf{diag}(x_i),\ x_i \in \mathbb{R}\ (i = 1, 2, \ldots, d)\}$ は $d \times d$ 対角行列全体の集合とする。$H \in \mathbb{S}^d_{++}$ に対して、H-内積と H-ノルムは、任意の $x, y \in \mathbb{R}^d$ に対して、それぞれ $\langle x, y\rangle_H := \langle x, Hy\rangle$, $\|x\|^2_H := \langle x, Hx\rangle$ と定義される。閉凸集合 X $(\subset \mathbb{R}^d)$ への距離射影は P_X と表し、H-ノルムにおける X 上の距離射影を $P_{X,H}$ と表す。また、確率変数 Y の期待値を $\mathbb{E}[Y]$ と表す。

2.2 仮定と問題

以下の仮定する。

(A1) $X \subset \mathbb{R}^d$ は射影が計算可能な閉凸集合である。
(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。但し、$\xi \in \Xi$ は確率分布 P に従う確率変数とする。$f : \mathbb{R}^d \to \mathbb{R}$ は、任意の $x \in \mathbb{R}^d$ に対して $f(x) := \mathbb{E}[F(x, \xi)]$ で定義する。

この仮定のもとで、関数 f の X 上での最小化問題の停留点 $x^\star$ を求めたい。すなわち、

$$x^\star \in X^\star := \{x^\star \in X^\star \colon \langle x - x^\star, \nabla f(x^\star)\rangle \geq 0\ (x \in X)\} \tag{1}$$

となる $x^\star$ を見つけたい。

以下の条件下での問題 (1) を考察する。

(C1) 独立同一分布に従う確率変数 $\xi_0, \xi_1, \ldots$ が存在する。
(C2) 入力点 $(x, \xi) \in \mathbb{R}^d \times \Xi$ は、$\mathbb{E}[\mathsf{G}(x, \xi)] = \nabla f(x)$ を満たす確率的勾配 $\mathsf{G}(x, \xi)$ を返す。
(C3) 正数 M が存在して、任意の $x \in X$ について、$\mathbb{E}[\|\mathsf{G}(x, \xi)\|^2] \leq M^2$ を満たす。

第 3 章 既存手法と提案手法

アルゴリズム 3.1[1] は、非凸最適化問題における停留点問題 (1) を解くための既存のアルゴリズムである。

アルゴリズム 3.1 における $(\mathsf{H}_n)_{n\in\mathbb{N}} \in \mathbb{S}^d_{++} := \mathsf{diag}(h_n, i)$ は以下を満たすとする。

(A3) 任意の $n \in \mathbb{N}, i = 1, 2, \ldots, d$ について、$h_{n+1,i} \geq h_{n,i}$ が成り立つ。
(A4) 任意の $i = 1, 2, \ldots, d$ について、$\sup\{\mathbb{E}[h_{n,i}] : n \in \mathbb{N}\} \leq B_i$ を満たす正数 B_i が存在する。
(A5) $x := (x_i) \in X$, $(x_n)_{n\in N} := ((x_{n,i}))_{n\in N}$ に対して、$D := \max_{i=1,2,\ldots,d} \sup\{(x_{n+1,i} - x_i)^2 \colon n \in \mathbb{N}\} < +\infty$ が成り立つ。

アルゴリズム **3.1** Adaptive learning rate optimization algorithm

入力: $(\alpha_n)_{n\in\mathbb{N}} \subset (0,1), (\beta_n)_{n\in\mathbb{N}} \subset [0,1), \bar{\beta} \in [0,1)$

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

loop

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

$\hat{m}_n := \frac{m_n}{1-\beta^{n+1}}$

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

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

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

$n \leftarrow n+1$

end loop

$\mathsf{H}_n \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$ と $v_n \in \mathbb{R}^d$ $(n \in \mathbb{N})$ を

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

と定義する。ただし、$v_{-1} = \hat{v}_{-1} = 0 \in \mathbb{R}^d$, $\hat{\beta} \in [0,1)$, $\eta > 0$ とし、

$$\mathsf{G}(x_n, \xi_n)^\eta := (\mathsf{G}(x_n, \xi_n)_i^\eta)_{i=1}^d$$

と定義する。$\eta = 2$ から成る (2) をもつアルゴリズム 3.1 は、Adaptive moment estimation (Adam) [2] と一致する。さらに、

$$\begin{aligned} v_n &:= \hat{\beta} v_{n-1} + (1-\hat{\beta})\mathsf{G}(x_n, \xi_n)^\eta \\ \hat{v}_n &:= (\hat{v}_{n,i}) := (\max\{\hat{v}_{n-1,i}, v_{n,i}\}) \\ \mathsf{H}_n &= \mathsf{diag}\left(\hat{v}_{n,i}^{\frac{1}{\eta}}\right) \end{aligned} \tag{3}$$

で定義するとき、$\eta = 2$ から成る (3) を有するアルゴリズム 3.1 は、Adaptive Mean Square Gradient (AMSGrad) [3] と一致する。

式 (2), (3) で定義された H_n と v_n は、(A5) のもとでは、(A3), (A4) を満たす（詳細については [1] を参照せよ）。

第 4 章　数値実験

明治大学所有の高速演算スカラーサーバーを利用した。2 つの Intel(R) Xeon(R) Gold 6148 CPU(2.4 GHz, 20 cores) 及び、NVIDIA Tesla V100 (16GB, 900Gbps) GPU、Red Hat Enterprise Linux 7.6 operating system を有している。実験プログラムは Python 3.8.2 を使用し、NumPy1.18.1 と PyTorch1.3.0 を使用した。

使用した学習データは 10 個のラベルが付いたラベル付けされたデータセット CIFAR-10 を使用した。計 6 万枚のうち、5 万枚の画像とラベル付けされたデータセットを訓練データとし学習させ、残りの 1 万枚のデータセットをテストデータとして精度を比較した。ニューラルネットワークモデルは畳み込み層が 44 層からなる ResNet44 を使用した。

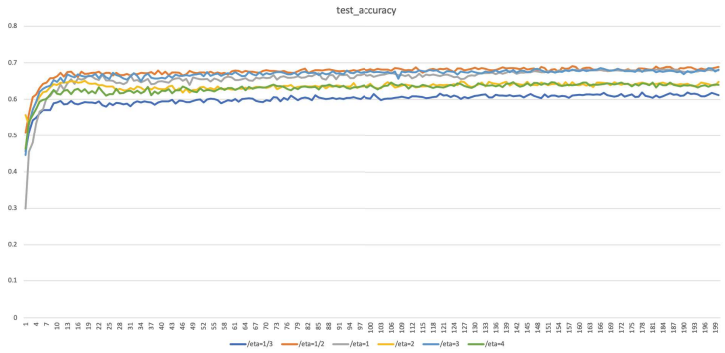

図 1　Adam (2), $\alpha = 10^{-3}$, $\beta = 10^{-3}$, $\eta = 1/3, 1/2, 1, 2, 3, 4$ を有するアルゴリズム 3.1 におけるテストデータの分類精度

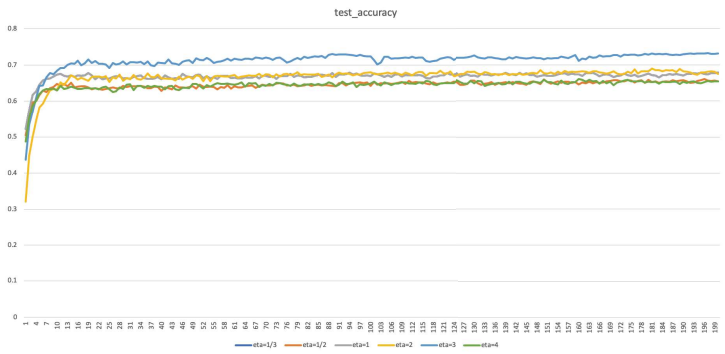

図 2　AMSGrad(3) , $\alpha = 10^{-3}$, $\beta = 10^{-3}$, $\eta = 1/3, 1/2, 1, 2, 3, 4$ を有するアルゴリズム 3.1 におけるテストデータの分類精度

実験結果は、既存の手法 ($\eta = 2$) より高い精度を得ることができることを示唆しているが、η はアルゴリズムを実装する前に設定する必要があるため、どのように η を設定するのが良いのか、検討する余地がある。

参考文献

[1] H. Iiduka. Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. 2021.

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

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