4S-08

確率的重み付け平均法のラージバッチ学習における有用性の検証

Verification of Generalization Performance Improvement using Stochastic Weight Averaging in Large Batch Training

所畑 貴大 *1　長沼 大樹 *1　横田 理央 *2
Takahiro Shohata　Hiroki Naganuma　Rio Yokota

*1東京工業大学 情報理工学院
School of Computing,
Tokyo Institute of Technology

*2東京工業大学 学術国際情報センター
Global Scientific Information and Computing Center,
Tokyo Institute of Technology

1. はじめに

近年, 深層学習を用いた手法は画像認識分野を始めとした多くの分野で活躍している. 一方, 深層ニューラルネットワークモデルの学習には膨大なパラメータやデータを用いるため, 学習時間が増加する傾向にあり, 学習の高速化が喫緊の課題である. しかし, 単純に一度に用いるデータ量を増やすことで高速化を図るラージバッチ学習では経験的に汎化性能が劣化するとされており, 原因としてスモールバッチ学習での Noise の影響が少なくなるため, 汎化性能の低い Sharp な解への収束へと陥る [1] という報告や, 目的関数の勾配の分散が小さくなることが原因で汎化性能が劣化するとされている [2]. 本研究では, モデルのパラメータを確率的に平均化する手法である確率的重み付け平均法 [3] をラージバッチ学習に適用することで, 汎化性能劣化問題への改善効果を検証する.

2. 背景

2.1 確率的勾配降下法

深層ニューラルネットワーク (DNN) の学習では出力 $y \in \mathbb{R}^K$ と期待される出力 $z \in \mathbb{R}^K$ との近さを測る損失関数 $L(y, z)$ を定義し, 損失関数を最小化するために反復的に DNN のパラメータ W を更新していく. L を最小化する方法として勾配降下法を元にした手法が一般に用いられる. 勾配 $\nabla L = \frac{\partial L}{\partial W}$ は L を増加させる方向を向いているため, W を $-\nabla L$ 方向へ更新することで L を減少させることができる. 勾配降下法では勾配を全訓練データを用いて計算するため多くの計算量と学習時間を要する.

ミニバッチ確率的勾配法 (Minibatch Stochastic Gradient Descent, 以下 SGD) では, すべてのデータを用いて計算される勾配ではなく, 訓練データから反復毎にランダムに複数取り出したデータ集合であるミニバッチを用いて計算される勾配を用いる. SGD は勾配降下法に比べ近似された勾配を用いることで計算量を削減できるため, 効率的な学習が期待でき広く用いられている.

2.2 大規模並列深層学習とラージバッチ学習

大規模並列深層学習には, DNN モデルのデータサイズが GPU などのアクセラレータのメモリでは不十分な場合などを解決するためのモデル並列と, 高速化を目的としたデータ並列の 2 つの並列方法が存在しており, 本研究ではデータ並列を主題としている. データ並列では全プロセスがモデルのパラメータの複製を保持しており, 各プロセスが異なるデータに対して計算を行いパラメータを更新していく手法である. この手法によりバッチサイズを大きくすることができ, 学習の高速化が可能となる. 一方, バッチサイズの増加に伴い汎化性能が劣化することが知られている. この問題の原因として大きく 2 つの仮説がある. 一つは同一 epoch 比較における学習のイテレーション数の低下による学習不足が原因とする説. もう一方は、小さなバッチ学習においては目的関数の勾配の分散が大きくなり汎化性能の高い最適解 (Flat Minima) に, ラージバッチ学習においては、目的関数の勾配の分散が小さいくなるため汎化性能の低い最適解 (Sharp Minima) に収束しやすい説 [2]. 本研究では後者の仮説に注目して汎化性能の改善を行った.

2.3 LARS

SGD では DNN の全ての層で同じ学習率を用いているが, 学習率 λ が大きい場合 $\| \lambda * \nabla L \|$ が $\| w \|$ より極端に大きくなり発散してしまう可能性がある. そこで, LARS(Layer-wise Adaptive Rate Scaling)[4] では各層ごとの重みと勾配の比率 $\frac{\|w\|}{\|\nabla L\|}$ に基づいて学習率を定めることで, 学習の発散を防ぎつつ大きな学習率を用いることを可能にする. LARS の詳しいアルゴリズムは Algorithm1 に示す.

Algorithm 1 LARS を用いたパラメータ更新

Parameters: γ: 学習率, m: モーメンタム, β: weight decay
Init: $v = 0$, 各層 l の重み w_0^l の初期化
  for $t = 1, \ldots, t_{max}$ **do**
    for $l = 1, \ldots, l_{max}$ **do**
      $g_t^l \leftarrow \nabla L(w_t^l)$ (ミニバッチの確率的勾配を取得)
      $\gamma_t \leftarrow \gamma$ (現在の学習率を取得)
      $\lambda^l \leftarrow \frac{\|w_t^l\|}{\|g_t^l\| + \beta \|w_t^l\|}$ (local LR の計算)
      $v_{t+1}^l \leftarrow m v_t^l + \gamma_t * \lambda^l * (g_t^l + \beta w_t^l)$
      $w_{t+1}^l \leftarrow w_t^l - v_{t+1}$
    end for
  end for

3. 確率的重み付け平均法

Izmailov らの提案した確率的重み付け平均法 (Stochastic Weight Averaging, 以下 SWA)[3] は, 学習終了直前のモデルの学習中にモデルのパラメータを定期的に抽出し, それらを平均化する手法である. SWA では式 1 の通りにパラメータ更新を行う.

$$W_{swa} = \frac{W_{swa} \cdot n_{models} + W}{n_{models} + 1} \tag{1}$$

ここで, n_{models} は今まで平均した重みの数である. SWA により, Flat Minima への収束が期待できる.

3.1 SWA のラージバッチ学習への適用

SWA をラージバッチ学習に活用した手法が Stochastic Weight Averaging in Parallel(以下 SWAP)[5] である. SWAP は並列深層学習を前提とした手法で, 最初に 150 エポックほ

連絡先: shohata@rio.gsic.titech.ac.jp

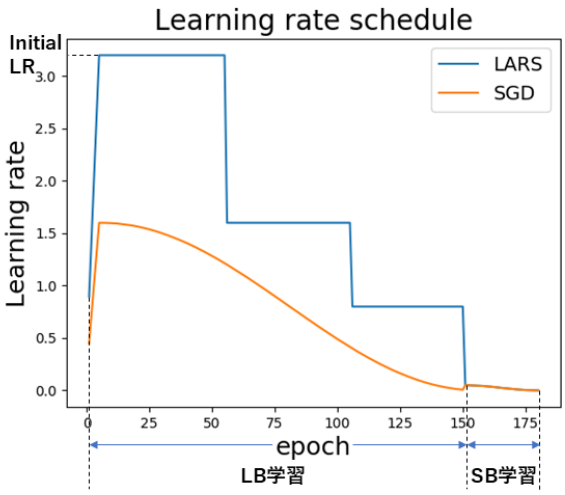
Copyright ©2020 Information Processing Society of Japan.
All Rights Reserved.

どラージバッチで学習した後, そこから複数の GPU で初期の W を共有した後, それぞれ独立した小さなバッチサイズでの学習を経て, それぞれ更新されたモデルを平均する. 学習の前半をラージバッチ学習 (LB 学習) により高速に行い, 後半の学習に SWA を並列化したスモールバッチ学習 (SB 学習) を適用することにより高い精度が期待できる. 本研究では, SWAP と LARS を組み合わせて使用することで, さらなる汎化性能の向上を目的とした.

4. 実験

実装は機械学習フレームワーク PyThoch[6] 上で行い, 画像分類タスクのデータセットとして CIFAR-10*1 を使用した. DNN モデルは ResNet18[7] を用いて学習を行った. オプティマイザーを SGD と LARS として, 各バッチサイズで SWAP による学習を行った. ただし後半の SB 学習では共通して SGD を使用した. 学習率スケジューリングは図 1 を用いた. 共通して学習の始めの 5epoch で Warmup[8] を行っている. また, LARS における initial LR はバッチサイズ 4096 の時の 3.2 を基準に, Linear Scaling Rule[8] に従ってバッチサイズに比例して増加させた.

図 1: 学習率スケジューリング



SGD と LARS による SWAP をそれぞれ 10 回実行した平均を比較した結果を表 1 に, LARS による SWAP の学習曲線を図 2 に示す. BS はバッチサイズを指す.

表 1: SGD, LARS の各 test accuracy

optimizer	test accuracy(%)		
	BS4096	BS8192	BS16384
SGD	93.22	92.48	85.79
LARS	94.60	94.07	91.67

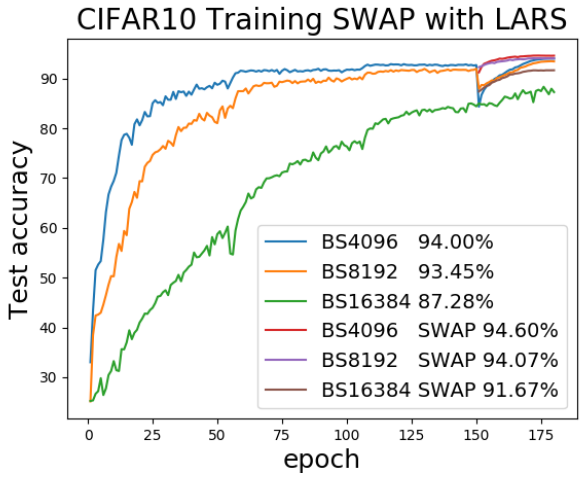
SGD を用いた場合と比較して, LARS を用いた場合の方が全てのバッチサイズにおいて汎化性能の改善が見られた. また BS16384 の結果から, SWAP への LARS の適用はラージバッチ学習において、既存手法より高い汎化性能改善の効果であることが確認できた.

5. おわりに

本研究では, LARS を用いた SWAP を提案した. SWAP の LB 学習時に LARS を用いることで, SGD を使用した場合と比較して汎化性能が改善することが確認できた. 今後の課題として, 小規模なデータセットでの検証だけではなく ImageNet などの大規模なデータセットでの検証が必要である.

*1 https://www.cs.toronto.edu/~kriz/cifar.html

図 2: LARS による SWAP の学習曲線



謝辞

本研究は，JST CREST JPMJCR19F5 の支援を受けたものである.

参考文献

[1] Smith S, Le Q. *A Bayesian Perspective on Generalization and Stochastic Gradient Descent.* ICLR 2018, 2018

[2] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, Ping Tak Peter Tang. *On large-batch training for deep learning: Generalization gap and sharp minima.* ICLR 2017, 2017.

[3] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. *Averaging weights leads to wider optima and better generalization.* UAI 2018, 2018

[4] Yang You, Igor Gitman, Boris Ginsburg. *Scaling SGD Batch Size to 32K for ImageNet Training.* 2017

[5] Vipul Gupta, Santiago Akle Serrano, Dennis DeCoste. *Stochastic Weight Averaging in Parallel: Large-Batch Training That Generalizes Well.* ICLR 2020, 2019

[6] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* NeurIPS 2019, 2019

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition.* CVPR 2016, 2016

[8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, Kaiming He. *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.* 2018

Copyright ©2020 Information Processing Society of Japan.
All Rights Reserved.