

大規模並列深層学習のための目的関数の平滑化 Smoothing of objective function for large scale parallel deep learning

長沼 大樹^{*1}
Hiroyuki Naganuma

横田 理央^{*2}
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
 ϵ は学習率と呼ばれ、更新の幅を決める係数である。

1. はじめに

深層学習では極めて冗長な数のパラメータを持つ深層ニューラルネットワーク (DNN) を膨大な学習データを用いて学習することで他の機械学習手法を圧倒する高い性能を発揮している。一方で、DNN の学習には膨大な計算時間がかかるため、大規模並列化によって学習時間を短縮するのが喫緊の課題である。SGD における小さなバッチサイズ (SB : Small Batch) での学習は確率的なノイズの影響で汎化性能の高い解 (Flat Minima) に収束する理論づけとして、SGD のプロセスが Random Potential の Random Walk に類似しているという説 [1] や、SGD はパラメータの近似ベイズ推定を行っているとみなせるため、SGD のノイズを調整をすることで良い汎化が期待できる [2] といった研究の報告がある。大きなバッチサイズ (LB : Large Batch) での学習ではそのノイズが適切ではなくなり、局所解から多少のパラメータ変動で誤差が極端に増加する Sharp Minima を避けることができず汎化性能が劣化する [3] といった報告や、Keskar らは、SGD を用いた DNN のバッチサイズを増加させた場合、目的関数の勾配が正確になり分散が小さくなることに起因して汎化性能が劣化している [4]。本研究では、汎化性能を改善するための前処理手法である Data Augmentation を、大きなバッチサイズ (LB : Large Batch) での学習に適用し、大規模並列化に伴うバッチサイズの増加により汎化性能が劣化する問題への解決手法となるか、また目的関数を平滑化する効果があるのかの検証を行う。

2. 背景

2.1 確率的勾配降下法 (SGD; Stochastic Gradient Descent)

DNN では入力 $x \in \mathbb{R}^d$ に対し出力 $y = f(x) \in \mathbb{R}^{d'}$ を計算する。DNN の学習においては出力 y と期待される出力 $z \in \mathbb{R}^{d'}$ との不一致度を測る損失関数 $L(y, z)$ を定義し、その期待値を目的関数とする最小化問題を解くために反復的にニューラルネットワークのパラメータ W を更新する。損失関数に関しては、多クラス分類を問題とする場合には交差エントロピーが広く用いられている。 W をユークリッド空間上の点と仮定すると、勾配 $\nabla L = \frac{\partial L}{\partial W}$ は L が増加する方向を向いている。したがって反復数 t でのパラメータ $w^{(t)}$ を勾配の逆方向、すなわち $-\nabla L$ 方向へ更新すれば L を減少させることができる。

SGD は、目的関数の勾配を直接用いず、1つのサンプルから近似してパラメータの更新を行うことで、反復回数1回の計算コストをはるかに小さくできることが利点である。この選択したサンプルでの損失関数を l とすると、更新式は以下の通りである。

$$w^{(t+1)} = w^{(t)} - \epsilon \nabla l(w^{(t)}) \quad (1)$$

2.2 ミニバッチ確率的勾配降下法 (Minibatch-SGD; Minibatch Stochastic Gradient Descent)

SGD を用いた場合、損失関数 $l(w^{(t)})$ はその期待値である目的関数と比べてずれているため目的関数の勾配方向からずれた向きに更新される。そのため、損失関数の目的関数からの誤差が大きいほど、勾配降下法と比べて反復回数に対する収束性能が劣化する。反復1回の計算コストを抑えながら、目的関数との誤差が小さい損失関数を扱うことで、収束速度を向上することが期待できる手法として、式2の通り、パラメータの更新を行うミニバッチ確率的勾配降下法が広く用いられている。

$$w^{(t+1)} = w^{(t)} - \epsilon \frac{1}{B} \sum_{l \in D_{(t)}} \nabla l(w^{(t)}) \quad (2)$$

ここで、 $D_{(t)}$ は、 B この学習データに対応する損失関数の集合で、ミニバッチと呼ぶ。各反復 t において、ミニバッチ $X^{(t)}$ は同時確率分布からサンプリングされると仮定すると、ミニバッチに対する損失関数 $\sum_{l \in D_{(t)}} \nabla l(w^{(t)})$ の期待値は目的関数と一致するので、1サンプルのみを用いるSGDと同じ解に収束する。一方で、ミニバッチ確率的勾配降下法を用いる場合の、損失関数の分散は $1/B$ となり、学習率 ϵ はバッチサイズなどを考慮して決める必要がある。一般的に、SGD とはミニバッチ確率的勾配降下法のことを指す場合が多く、本論文ではこの意味でSGDと表記する。

2.3 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おける並列方法には大きく分けて、モデル並列とデータ並列の2つの方法がある。モデル並列は、GPUなどのアクセラレータのメモリ不足などの対応のための文脈で研究が進められ、モデルのパラメータ w を複数のプロセス間で分散保持し、順伝播や逆伝播の際に通信を行う並列手法である。もう一方の並列方法であるデータ並列では、全プロセスがモデルのパラメータ w のコピーを保持し、異なるデータに対して各プロセスが計算を行い、定期的に同期を行う手法である。データ並列のアプローチによってバッチサイズを大きくし、目的関数との誤差が小さい損失関数を扱うことで、DNNの学習時間の短縮を行うことが期待されるが、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おり [5]、この精度劣化への対策に関する研究が近年急速に進んでいる。

3. Data Augmentation (DA)

DNN を用いた一般的な画像処理の研究において、既存の学習データに対し変形・ノイズ付加することで学習データを拡張させる Data Augmentation (DA) と呼ばれる手法が提案されており、学習画像に様々な処理を行うことで、汎化性能の高い DNN が学習できることが知られている。

3.1 Mixup

Hongyi らの提案した Mixup [7] は別のクラスの2つのラベル・データ双方を線形補完しある比率で新たなデータを作成する手法であり、ドメイン知識なしでも DA が可能である。 x をデータ、 y を onehot 表現のラベルとし、 $(x_i, y_i), (x_j, y_j)$ をランダムに選ばれたサンプル、 $\lambda \in [0, 1], \lambda \sim \text{Beta}(\alpha, \alpha)$ を仮定し、

式3の通り、 $(\tilde{x}, \tilde{y})$ というデータラベルのペアを作成する。

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda) x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda) y_j\end{aligned}\quad (3)$$

Mixup をはじめとする DA 手法は、LB 学習における汎化性能の改善のためのみに開発されたわけではない一方で、LB 学習において問題となっている減少するノイズや分散への解決策として、目的関数の平滑化の役割を果たすことで、汎化性能の改善が可能かどうかを検証する。

4. 実験

画像認識分野においてベンチマークとして広く用いられる、CIFAR-10^{*1} を学習データとしてを用いた。この学習データは、第3章で紹介した DA の一種である Mixup を用いた前処理をそれぞれ行ない、バッチサイズを 128(SB) と 8192(LB) として、DNN モデルとしては VGG9 を用いて学習を行った。また実装には pytorch^{*2} を用いており、損失関数の可視化は Hao[8] を参考に行った。

実験 1: 各種 Data Augmentation 手法による損失関数の可視化実験

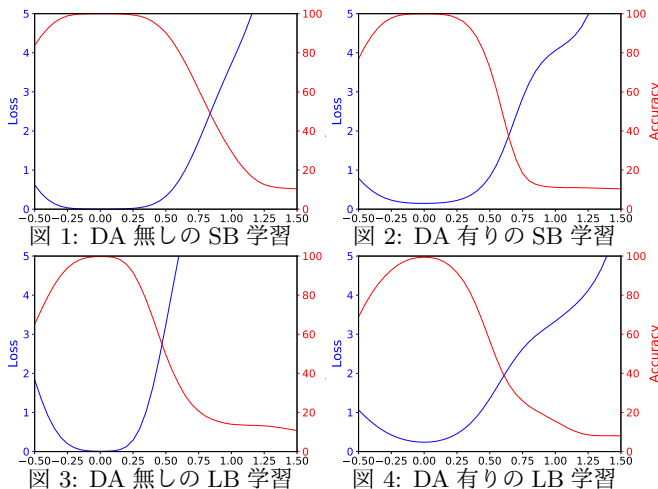


図 1: DA 無しの SB 学習

図 2: DA 有りの SB 学習

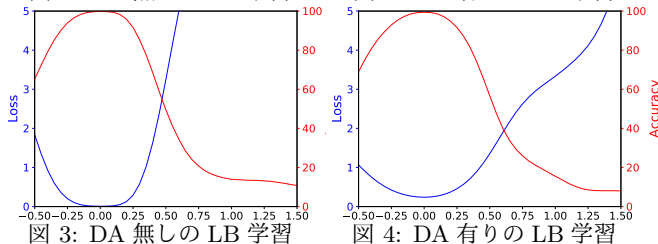


図 3: DA 無しの LB 学習

図 4: DA 有りの LB 学習

図 5: VGG9 学習で得られた解の 1 次元線形補完図。青い線は損失値、赤い線はの認識精度を示す。

異なるバッチサイズにおいて、DA を適用させ、その際の損失関数の変化を図 5 の通り可視化にて確認した。特に LB 学習では、Sharp Minima の形状が改善する結果となった。

実験 2: 各種 Data Augmentation 手法による汎化性能の改善実験

DA を用いた学習における特定のバッチサイズでの学習曲線を図 6 に示す。DA を用いない場合に比べ、SB/LB の学習共に汎化性能の改善が見られた。

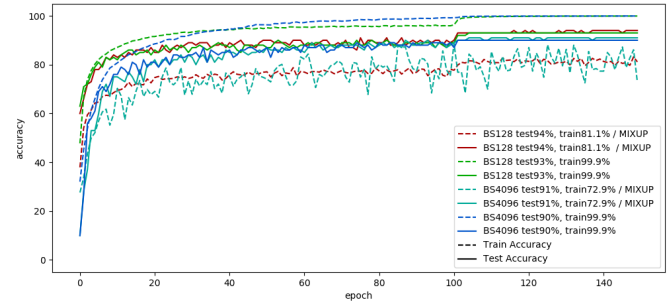


図 6: DA 手法による SB/LB での学習曲線、VGG9 に対して異なるバッチサイズでの学習に対し DA を行う場合と行わない場合の学習曲線を比較した、BS はバッチサイズを、test, train は test データでの評価、train データでの評価をそれぞれ表す

5. おわりに

本研究では LB 学習に DA を用いることで、SB 学習と同様に汎化性能の改善が達成できると共に、特に LB 学習においては DA を適用することによる目的関数の平滑化の効果が確認できた。一方で、今回の可視化に関しては、軸の定め方などの議論の余地があり、ラージバッチ学習における汎化性能の劣化と目的関数の平滑性の関連について明確な説明が重要である、また今回の実験においては VGG9 という比較的浅い DNN で学習を行ったが、より深い DNN では目的関数の滑らかさが変化するため、検証が必要である。

謝辞

本研究は、JST CREST JPMJCR1687 の支援を受けたものである。

参考文献

- [1] Hoffer E, Hubara I, Soudry D: *Train longer, generalize better: closing the generalization gap in large batch training of neural networks* NIPS, 2017.
- [2] Mandt S, Hoffman M, Blei D: *Stochastic Gradient Descent as Approximate Bayesian Inference*, Journal of Machine Learning Research 18 (2017) 1-35, 2017.
- [3] Smith S, Le Q: *A Bayesian Perspective on Generalization and Stochastic Gradient Descent*, ICLR 2018, 2018.
- [4] Keskar N, Mudigere D, Nocedal J, Smelyanskiy M, Tang P: *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima*, ICLR 2017, 2017.
- [5] Yang Y, Igor G, Boris G: *Large Batch Training of Convolutional Networks*, arXiv:1708.03888, 2017.
- [6] Zhun Z, Liang Z, Guoliang K, Shaozi L, Yi Y: *Random Erasing Data Augmentation*, arXiv:1708.04896, 2017.
- [7] Hongyi Z, Moustapha C, Yann N D, David L: *mixup: Beyond Empirical Risk Minimization*, ICLR 2018, 2018.
- [8] Hao L, Zheng X, Gavin T, Christoph S, Tom G: *Visualizing the Loss Landscape of Neural Nets*, Neural Information Processing Systems, 2018.

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

*2 <https://pytorch.org/>