4R-08

深層ニューラルネットワークにおける ヘッセ行列の近似手法による固有値への影響調査

Measuring the effect of approximation methods on the eigenvalues of Hessian in DNN

本川 哲哉 *1　長沼 大樹 *2　井手 達郎 *3
Tetsuya Motokawa　Hiroki Naganuma　Tatsuro Ide

*1筑波大学大学院 図書館情報メディア研究科
Graduate School of Library Information and Media Studies, University of Tsukuba

*2モントリオール大学, Mila
Université de Montréal, Mila - Quebec Artificial Intelligence Institute

*3カリフォルニア大学アーバイン校
University of California, Irvine

深層学習の汎化性能にヘッセ行列の固有値が強く相関を持つことが知られている．近年，最先端の性能を発揮する学習手法も，ヘッセ行列の固有値を考慮した方法となっている．しかしながら，深層 NN では計算量が膨大となるため，一般に近似手法によって計算されたヘッセ行列が用いられている．本研究では，深層学習において，各種近似手法がヘッセ行列の固有値に及ぼす影響を調査する．

1. はじめに

近年，ニューラルネットワーク（以下 NN）の学習ダイナミクスの理解や最適化効率の改善のために，損失関数のヘッセ行列を利用する研究が盛んに行われている．通常 NN の学習では，損失関数の一次微分である勾配を用いて学習が行われる．一次の勾配法は停留点に収束するため，学習後半は勾配のノルムはほぼ 0 に収束する．そのため，パラメータの変化に対する勾配の変化率を表すヘッセ行列で収束先のパラメータを定量化し，分析することは自然な方法である．一方でパラメータ数の膨大な深層 NN（Deep Neural Network，以下 DNN）ではヘッセ行列の計算が困難である．これまで様々な方法で DNN における効率的なヘッセ行列計算手法の研究・開発が行われてきた．本研究では，特に各近似手法がヘッセ行列の固有値に及ぼす影響を実験的に調査した．

2. 背景

2.1 ヘッセ行列計算の効率化に関する研究

大規模な DNN モデルにおけるヘッセ行列を Exact に計算するのは空間計算量・時間計算量共に現実的には困難である．

2.1.1 Hessian vector propduct の高速な計算

深層学習における最適化においては，ヘッセ行列そのものではなく，ヘッセ行列と任意のベクトルとの積（Hessian vector product，以下 Hvp）さえ計算出来れば十分であり，Hvp を Exact に計算するために，Pearlmutter（1994）は NN における Hvp を誤差逆伝播中に高速に計算するアルゴリズムを提案した [4]．この Hvp を応用し，最適化ではない用途，例えばヘッセ行列の最大固有値を求めたい場合，べき乗法（power iteration）によって近似的に計算が可能である．べき乗法ではヘッセ行列 $\boldsymbol{H}$ の最大固有値を求めたいときに，適当なベクトル $\boldsymbol{v_0}$ を初期値として，逐次的に $\boldsymbol{v_{k+1}} = \boldsymbol{H v_k}$ の計算を繰り返す．$\boldsymbol{v_k}$ が行列 $\boldsymbol{H}$ の最大固有ベクトルの方向に収束していくことを利用して最大固有値を計算する．

また，ヘッセ行列のトレースの近似計算には Hutchinson Method として知られるアルゴリズムがよく用いられる [1]．Hutchinson Method ではヘッセ行列とラデマッハランダムベクトル（各要素が $\frac{1}{2}$ の確率で 1 or -1 を取る）との二次形式の期待値を近似的に計算する．

$$\begin{aligned} tr(\boldsymbol{H}) &= tr(\boldsymbol{HI}) = tr(\boldsymbol{H}E[\boldsymbol{vv}^T]) \\ &= E[tr(\boldsymbol{Hvv}^T)] = E[\boldsymbol{v}^T\boldsymbol{Hv}] \end{aligned} \tag{1}$$

2.1.2 The Generalized Gauss-Newton Matrix

本節では，DNN のヘッセ行列の近似行列としてよく利用される The Generalized Gauss-Newton Matrix（以下 GGN 行列）について議論する [5]．古典的には $\tilde{\boldsymbol{G}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{J_\theta})^T\boldsymbol{J_\theta}$ で定義される近似ヘッセ行列を Gauss-Newton 行列と呼び，準二次最適化手法の中で広く使用されてきた．ここで，$\boldsymbol{J_\theta}$ は NN 出力 $f(\boldsymbol{\theta}; \boldsymbol{x}_i, \boldsymbol{y}_i)$ のパラメータに関するヤコビ行列を表す．

Schraudolph は Gauss-Newton 行列をより一般的に拡張し，$\boldsymbol{G}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{J_\theta})^T\boldsymbol{H}_L\boldsymbol{J_\theta}$ と定義した [5]．$\boldsymbol{H}_L$ は出力層のみのパラメータにおけるヘッセ行列である．特に $\boldsymbol{H}_L$ が単位行列として表せる場合，古典的な Gauss-Newton 行列 $\tilde{\boldsymbol{G}}$ に一致する．また活性化関数として ReLU を使用する場合，その二次微分の値が 0 になることを利用してヘッセ行列と GGN 行列が一致することが知られている [2]．現実的な DNN の設定では活性化関数として ReLU を用いることが多いため，GGN 行列を計算すれば十分であることがわかる．

3. 実験

MNIST 学習後の NN に対して，Exact なヘッセ行列の固有値と近似的に計算された固有値の近似性能の比較を行った．Exact なヘッセ行列を計算するためにモデルにはパラメータ数の少ない NN を用いた．

3.1 ヘッセ行列計算のためのライブラリ

本研究では 2 章で説明したようなヘッセ行列の近似計算手法が効率的に実装されたライブラリを用いて実験を行った．また，Exact なヘッセ行列は Pytorch[*1] の自動微分を二回適用することで陽に二次微分を計算して求めた．

3.1.1 PyHessian

PyHessian[*2] は Yao ら（2020）によって開発された Pytorch と互換性のあるライブラリである [6]．Hvp を用いて効率的に固有値やトレースの計算が可能となっている．

連絡先: moto.t.03.td@gmail.com

∗1 https://pytorch.org
∗2 https://github.com/amirgholami/PyHessian

3.1.2 BacPACK

BackPACK[*3] は Dangel ら（2020）によって開発されたヘッセ行列の近似行列を計算できる Pytorch 上で動くライブラリである [3]．GGN 行列や対角ヘッセ行列を計算するために誤差逆伝播中に必要な情報を計算・保持するように設計されている．

3.2 上位 10 固有値分布の近似性能比較

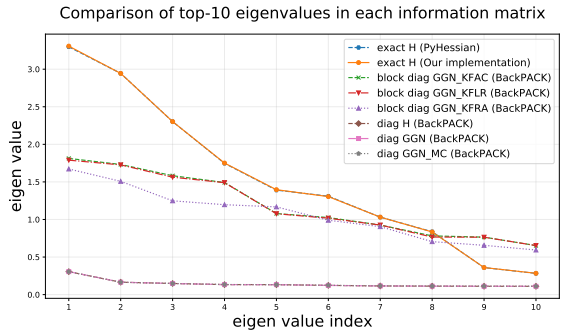


図 1: Exact に計算したヘッセ行列の上位 10 固有値と近似計算した上位 10 固有値の比較．

図 1 より PyHessian によって計算された固有値と Exact に陽な形で計算されたヘッセ行列の固有値との差はほぼない．べき乗法によってほぼ Exact な固有値に収束しているということがわかる．それに対して BackPACK によって計算された近似行列の固有値とは乖離がある．特に対角近似を行う 3 手法全てにおいて，Exact な固有値とは大きく異なる．これは Exact なヘッセ行列の全ての非対角成分が正である場合，対角近似を行うと最大固有値が小さくなるためであると考えられる．

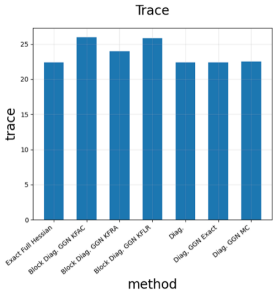
3.3 トレースの近似性能比較

図 2 より，対角近似行列のトレースは Exact なヘッセ行列のトレースと一致しており，対角成分が正確に計算できていることがわかる．GGN 行列のブロック近似の 3 手法に関しては全てわずかに Exact なトレースとは異なる値を取っており，これはクロネッカー因子分解に起因するものと考えられる．また，PyHessian の Hutchinson Method によるトレースの近似計算の結果は Exact なトレースとわずかに乖離している．これは Hutchinson Method では期待値計算（式 1）に用いる二次形式をサンプリング近似するためである．そこでサンプルサイズがトレースの推定値に与える影響を調査した（図 3）．サンプルサイズが大きいほど Exact なトレースの値（22.40）に近づいていくことがわかる．

4. おわりに

本研究ではヘッセ行列の固有値に関する近似性能評価を行った．特に BackPACK と PyHessian の 2 つのライブラリに対して，Exact なヘッセ行列を計算する実装を行い，固有値とトレースの近似精度に関する比較実験を行った．結果として ReLU を用いた NN の分類タスクにおいて，BackPACK で計算できる対角近似行列のトレースと PyHessian の上位 10 個の固有値は Exact なヘッセ行列と結果が一致することを確認した．また，PyHessian のトレースに関してサンプルサイズを大きくすることで Exact なトレースに近づいていくことも確認できた．

*3 https://backpack.pt



Trace and Maximum Eigenvalue of Hessian and Their Elapsed Times

	Full Hessian	PyHESSIAN
Tr(H)	22.40	19.98
Elapsed: Tr(H)	10.18 sec.	7.64 sec.
Maximum Eigenvalue	3.31	3.29
Elapsed: All Eigs.	32.26 sec.	20 hr.

図 2: Exact に計算したヘッセ行列のトレースと近似計算したトレースの比較（左：BackPACK との比較，右；PyHessian との比較（サンプルサイズ 25））

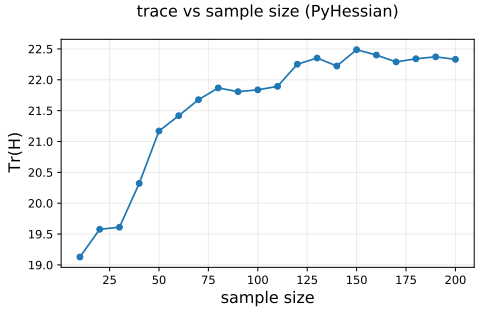


図 3: Hutchinson Method によるサンプルサイズがトレースの推定値に与える影響

参考文献

[1] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58(2), 2011.

[2] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 557–565, 2017.

[3] Felix Dangel, Frederik Kunstner, and Philipp Hennig. BackPACK: Packing more into backprop. In *International Conference on Learning Representations*, 2020.

[4] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6:147–160, 1994.

[5] Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002.

[6] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *ICML workshop on Beyond First-Order Optimization Methods in Machine Learning*, 2020.