2V-06

フィッシャー情報行列の高速近似計算の固有値解法への適用

石井 央† 横田 理央‡

†東京工業大学 情報理工学院 ‡東京工業大学 学術情報国際センター

1. はじめに

深層学習において，ニューラルネットワークの loss landscape の曲率情報など幾何学的情報を調べるにあたって，フィッシャー情報行列やヘッセ行列などの2次勾配情報行列が重要な役割を果たす．これらの情報行列の固有値計算は，パラメータ数が多くなるほど実用的ではなくなる．本研究では，高速近似計算で求めたフィッシャー情報行列の固有値密度を Stochastic Lanczos Quadrature [1]と呼ばれるアルゴリズムで近似計算し，正確なフィッシャー行列，固有値密度と比較する．

2. フィッシャー情報行列の近似計算

2.1 フィッシャー情報行列

パラメータベクトル$\boldsymbol{\theta} \in \mathbb{R}^{\mathrm{n}}$に対するフィッシャー情報行列$\boldsymbol{F_\theta} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$は以下の式で定義される．

$$\boldsymbol{F_\theta} = \mathbb{E}[\boldsymbol{\nabla_\theta} \log f(\boldsymbol{X};\boldsymbol{\theta}) \boldsymbol{\nabla_\theta} \log f(\boldsymbol{X};\boldsymbol{\theta})^{\mathrm{T}}] \quad (1)$$

ここで，$f(\mathbf{X};\boldsymbol{\theta})$は$\boldsymbol{\theta}$によって定まる$\boldsymbol{X}$の確率密度関数であり，期待値は$\boldsymbol{X}$に対して取られる．ニューラルネットワークにおいて，$f(\mathbf{X};\boldsymbol{\theta})$はソフトマックス関数の出力，すなわちネットワークの出力$p(k|\boldsymbol{x};\boldsymbol{\theta})$に相当する．フィッシャー情報行列には多くの近似手法がある[2]．その中で，本研究で用いたものを説明する．

2.2 モンテカルロ近似

式(1)の期待値を計算するためには，クラスの数だけ繰り返し計算が必要になる．ただ，CIFAR-100 や ImageNet ではクラス数が 100，1000 となり，計算コストが高くなってしまう．モンテカルロ近似を用いたフィッシャー情報行列の計算では，モンテカルロ法により抽出されたmクラスのみで計算を行う．このmをクラス数よりも小さくすることで計算コストを抑えることができる．

2.3 エンピリカルフィッシャー

エンピリカルフィッシャーは，式(1)において$f(\mathbf{X};\boldsymbol{\theta})$に，ネットワークの出力ではなく学習用のデータ$(\boldsymbol{x},\boldsymbol{y})$の分布$p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$を用いて計算される行列のことを指す[3]．エンピリカルフィッシャーの計算ではクラスごとのループ計算が必要ないので，正確なフィッシャー情報行列を計算するよりも計算コストは抑えられる．

2.4 ブロック対角近似

モンテカルロ近似やエンピリカルフィッシャーは行列を定義する段階での近似であるが，ブロック対角近似は行列の計算をする際の近似である．ブロック対角近似では，ニューラルネットワークの異なる層のパラメータ間に相関がないという近似をして計算を行う．

3. 実験

3.1 固有値密度の計算

ある行列の固有値のスペクトル密度は以下の式のように表せる．

$$\phi(t) = \frac{1}{m}\sum_{i=1}^{m} \delta(t-\lambda_i) \quad (2)$$

ここで，δはディラックのデルタ関数でλ_iは行列の固有値である．固有値を全て求めれば計算は簡単だが，情報行列などのように行列が大きくなると実用的ではない．行列のスペクトルを含む区間をより小さな区間に分け，それぞれの区間の固有値の数を求めて$\phi(t)$を近似する方法も考えられる．それぞれの区間の幅を小さくしていけば，この近似は行列のスペクトル密度に近づくが，これも計算コストが高い．ここでは，行列積を用いて計算する Stochastic Lanczos Quadrature [1] (SLQ) と呼ばれるアルゴリズムを使用する．原理を簡単に説明すると，まずディラックのデルタ関数をガウス関数で近似する．このガウス関数で置き換えられた式は，ガウス求積法などを用いると，固有値と固有ベクトルの第一要素で表される近似式にできる．SLQ は，この固有値と固有ベクトルをランチョス法[4]により求めて固有値密度の近似式を計算するアルゴリズムである．

3.2 データセットとモデル

データセットは MNIST と CIFAR-10 の学習用のデータを使用した．モデルは図1の CNN を用いた．

Application of fast approximate calculation of Fisher Information Matrix to eigenvalue algorithm
Hiro Ishii† Rio Yokota‡
†School of Computing, Tokyo Institute of Technology
‡Global Scientific Informaiton and Computing Center, Tokyo Institute of Technology

図 1：使用した CNN のアーキテクチャ

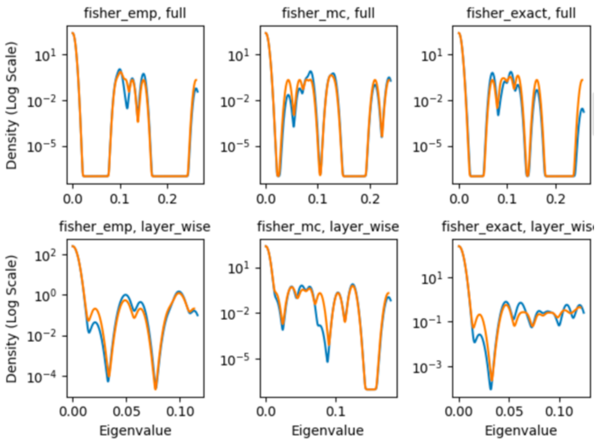

図 2：MNIST による実験結果

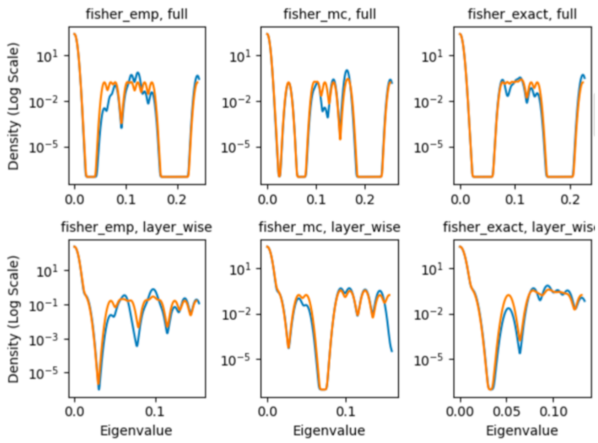

図 3：CIFAR-10 による実験結果

MatMul 層と Gemm 層は全結合層を示す．最初の Conv 層と MatMul 層の重みの形はデータセットごとに変更した（図は CIFAR-10 用のアーキテクチャ）．

3.3 行列近似

MNIST と CIFAR-10 による実験結果をそれぞれ図 2 と図 3 に示す．近似手法の組み合わせによりそれぞれ 6 つのプロットがある．fisher_exact が実際のフィッシャー情報行列，fisher_mc が $m = 1$ でモンテカルロ近似したフィッシャー情報行列，fisher_emp がエンピリカルフィッシャーとなっている．また，full は行列の全ての要素を求めたことを表し，layer_wise がブロック対角近似を行なったことを表す．それぞれのプロットにおいて，横軸は固有値，縦軸はログスケールの固有値密度となっている．橙色のプロットは実際の固有値から求めた固有値密度で，青色のプロットは SLQ から求めた固有値密度となっている．実際の固有値を求めてガウス関数への入力とし，出力されたものが橙色のプロットである．実装には ASDL[*1] を用いた．

4. 結論

図 2，図 3 を見ると，SLQ により，実際の固有値から求められた固有値密度と同じような固有値密度を求められることがわかる．また，fisher_exact と fisher_mc，fisher_emp を比べると，fisher_emp，fisher_mc どちらでも fisher_exact の固有値密度の形と重なる部分があることがわかる．full と layer_wise を比べると，layer_wise において固有値の範囲が小さくなっているが，その範囲では full の固有値密度のおおよその形を示している．よって，これらの近似をしても，ある程度は実際の固有値密度の形が反映されているため有意な結果が得られることが確認できた．今後の課題として，実用場面で使われるようなより大きなモデルでの検証が必要である．

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] Zhewei Yao, Amir Gholami, Kurt Keutzer, Michael W. Mahoney. PYHESSIAN: Neural Networks Through the Lens of the Hessian. arXiv preprint arXiv:1912.07145, 2020.
[2] Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Akira Naruse, Rio Yokota, Satoshi Matsuoka. Large-Scale Distributed Second-Order Optimization Using Kronecker-Factored Approximate Curvature for Deep Convolutional Neural Networks, IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2019.
[3] James Martens. New insights and perspectives on the natural gradient method. arXiv preprint arXiv:1412.1193, 2017.
[4] Gene H Golub and Gérard Meurant. *Matrices, moments and quadrature with applications.* Princeton University Press, 2009.

*1 https://github.com/kazukiosawa/asdfghjkl