1A2-OS-05b-4

自然勾配近似法を起点としたバッチ正規化の数理的理解

Mathematical Understanding of Batch Normalization as Approximation of Natural Gradient

木脇太一 *1
Taichi Kiwaki

*1東京大学 情報理工学系研究科
Schoold of Information Science and Technology, The University of Tokyo

Batch normalization is a widely accepted technique for training a deep neural network because batch normalization drastically speeds up the learning process of a neural network. A connection between batch nomalization and natural gradient has been suggested from qualitative and experimental point of views. However, mathematical mechanism behind batch normalization remains unrevealed. In this manuscript, I analyze the effect of batch normalization and depict the relationship toward natural gradient under several conditions and approximations.

1. はじめに

近年、画像分類問題への成功を契機として深層ニューラルネットワークの研究が活発に行われている [8, 13, 21, 22]。その中でニューラルネットワークの学習効率化をはかる手法として、近年提案されたものにバッチ正規化がある [10]。BN は単純ながら劇的な学習高速化を実現するため広く利用が進んでいる [8, 9, 11, 15, 17, 20, 23]。ではどのような原理をもってバッチ正規化はニューラルネットワークの学習を助けるのであろうか？実験的および定性的な観察からの仮説として *1、バッチ正規化が自然勾配法 [1] を近似的に実現することが理由であると言われているが [19]、数理的な詳細は未だ不明瞭である。また動作原理の不明瞭さに関連して、BN を導入する箇所に関しても議論の余地が残されている [15]。

これらの議論を背景として、本稿ではバッチ正規化と自然勾配法の関係性に関して数理的な分析を行う。その結果として、いくつかの条件および近似のもとでバッチ正規化は隣接する階層のパラメータ間における自然勾配法を実現することを示す。また併せてバッチ正規化を導入する位置とその効果に関しても考察する。

2. MLP

以下では次の様な L 層パーセプトロン（Multi Layered Perceptro: MLP）を考える。

$$\begin{aligned} p(y=1|\mathbf{x}) = \mathbf{h}^{(L)} &= f^{(L)}(\mathbf{a}^{(L)}) \\ \mathbf{a}^{(L)} &= W^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)} \\ &\cdots \\ \mathbf{h}^{(1)} &= f^{(1)}(\mathbf{a}^{(1)}) \\ \mathbf{a}^{(1)} &= W^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \end{aligned} \tag{1}$$

ここで $\mathbf{x} \in \mathbb{R}^{N_{\text{in}}}$ は説明変数、$y \in \{0, 1\}$ は説明変数である。$(W^{(l)}, \mathbf{b}^{(l)})$ は第 l 層におけるネットワークのパラメータ、$f^{(l)} \in \mathbb{R} \to \mathbb{R}$ は（一般的に非線形の）活性化関数、$\mathbf{a}^{(l)} \in \mathbb{R}^{N^{(l)}}$ は第 l 層のニューロン数を $N^{(l)}$ とした活性化関数への引数、そして $\mathbf{h}^{(l)} \in N^{(l)}$ はその出力である。また 1 変数関数 $f(x)$ の引数に多次元ベクトル $\mathbf{x} = (x_1, \ldots, x_N)^\top$ を渡した際には、暗黙に $f(\mathbf{x}) = (f(x_1), \ldots, f(x_N))^\top$ と $\mathbb{R}^N \to \mathbb{R}^N$ への拡張を行うとした。

MLP の学習は次の様に学習損失 $L(\theta)$ の最小化問題として定式化される。

$$\theta^* = \text{argmin}_\theta \, L(\theta) \tag{2}$$

$$L(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim\pi} \left[-\log p(y|\mathbf{x}; \theta) \right] \tag{3}$$

この問題を実際に解くためには確率的勾配降下法 (Stochastic Gradient Descent: SGD) が広く利用される。SGD ではミニバッチと呼ばれる $(y, \mathbf{x})$ の少数サンプルから損失に対する θ の勾配を推定して逐次的にパラメータを更新する。

3. バッチ正規化

バッチ正規化 [10] は MLP のある変数 a に対して、次の様な線形変換として定義される。

$$\text{BN}\,(a) = \frac{a - \mu_{\text{B}}}{\sigma_{\text{B}}} \tag{4}$$

ここで μ_{B} および σ_{B} はミニバッチから算出した a の平均および標準偏差である。

バッチ正規化を導入する箇所としては非線形性の前 $\mathbf{a}^{(l)}$ および後 $\mathbf{h}^{(l)}$ が報告されているが、本稿では特に断らない限りより一般的である $\mathbf{a}^{(l)}$ に対するバッチ正規化を考える。この場合、$\tilde{\mathbf{a}}^{(l)} = \text{BN}\left(\mathbf{a}^{(l)}\right)$ を $\mathbf{a}^{(l)}$ に替わって $f^{(l)}$ の引数とする。バッチ正規化は線形変換であるため、$\mathbf{h}^{(l)}$ から $\mathbf{a}^{(l)}$ への変換と併せて

$$\tilde{\mathbf{a}}^{(l)} = \text{BN}\left(W^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}\right) = \tilde{W}^{(l)}\mathbf{h}^{(l)} + \tilde{\mathbf{b}}^{(l)} \tag{5}$$

の様に取り扱うことができる。

バッチ正規化において平均と標準偏差に計算が簡単なミニバッチ統計量を利用するのは計算上の都合である [10]。よってこの点はバッチ正規化の数理的な意義を考慮する上では本質的で無いと考え、以下の分析ではバッチ正規化と言いつつも、暗に μ_{B} および σ_{B} は a の真の平均および標準偏差を利用していると理想化して議論する。

連絡先: 氏名，所属，住所，電話番号，Fax 番号，電子メールアドレスなど

*1 Salimans and Kingma [19] では Ioffe and Szegedy [10] が議論しているとしているが、実際に Ioffe and Szegedy [10] には記述が無いため正確な理由については不明である。

4. 自然勾配法

SGD の各ステップにおけるパラメータ更新は次の様な最適化問題を解いていると理解ができる。

$$\theta^{(t+1)} = \mathrm{argmin}_\theta \left\{ \left\langle \frac{\partial L}{\partial \theta}, \theta - \theta^{(t)} \right\rangle - \left\| \theta - \theta^{(t)} \right\|_2^2 \right\} \quad (6)$$

ただし $\frac{\partial L}{\partial \theta} = (\frac{\partial L}{\partial \theta_1}, \ldots, \frac{\partial L}{\partial \theta_N})^\top$ と定義した。これはパラメータの空間に対してユークリッド距離を導入し、その距離を一定とする制限の元で最も損失を下げる方向を見つけ出していることに他ならない。しかしながら統計的モデルのパラメータの様な抽象的な対称に対して、ユークリッド距離が適切な計量であるとは限らない。Amari [1] は統計的モデルの集合をリーマン多様体と見なし、フィッシャー情報量行列

$$F_\theta = \mathbb{E}_{\mathbf{x}\sim\pi} \left\{ \mathbb{E}_{y\sim p(y|\mathbf{x};\theta)} \left[\left(\frac{\partial \log p}{\partial \theta} \right) \left(\frac{\partial \log p}{\partial \theta} \right)^\top \right] \right\} \quad (7)$$

により定まるリーマン計量を距離の指標として採用する自然勾配法を提案した。自然勾配法を利用することにより、ニューラルネットワークの学習における特異点付近での学習停滞現象が解消されうることが知られている [2, 4]。

自然勾配法の実現のためには大きく分けて二つの方法がある。一つ目の方法は、明示的にフィッシャー情報量行列およびその逆行列を計算し自然勾配を求める方法である [1, 6, 7, 12, 14, 18]。このうち最も基本的なものは Amari [1] の直接フィッシャー情報量行列を推定する方法である。しかしながら大規模ネットワークにおけるフィッシャー情報量行列は莫大な大きさとなる。計算上の負担を軽減するために、フィッシャー情報量行列の対角近似 [12][*2]、低ランク近似 [18]、スパース行列近似 [7]、クロネッカー因子分解 [6, 14] などに基づくアルゴリズムが提案されている。

二つ目の方法は、明示的にフィッシャー情報量行列を取り扱うのではなく、ネットワークのパラメータの取り方を変えることなどによりフィッシャー情報量行列を単位行列に近づける方法である [3, 5, 16, 19]。これにはパラメータや確率変数の座標変換に対するネットワークの普遍性の除去 [3, 16, 19] や、パラメータ変換によるフィッシャー情報量行列の対角化 [5] が知られている。

次の章では数理的な分析を通して、ある条件のもとでバッチ正規化が上記のうち二つ目の自然勾配の近似法として機能することを示す。

5. バッチ正規化の分析

以下では分析の見通しを良くするため、全ての層でニューロン数は一定数 N であるとする。また $\tilde{W}^{(l)}$ は正則行列であるとする。

ここで着目する量は、隣接階層のパラメータ間に対応するフィッシャー情報量行列の部分行列、

$$F_{l,l-1} = \mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\mathbf{vec}\left(\delta^{(l)} \mathbf{h}^{(l)\top} \right) \mathbf{vec}\left(\delta^{(l-1)} \mathbf{h}^{(l-1)\top} \right)^\top \right] \quad (8)$$

である。ここで $\delta^{(l)}$ は l 層における逆伝搬信号である。これはフィッシャー情報量行列の非対角部分であるから、自然勾配を実現するためにはこれがゼロとなれば良い。この部分行列は $\delta^{(l)}$ と $\mathbf{a}^{(l)}$ の統計的独立性を仮定することで、次のクロネッカー因子分解 [5, 6, 14]

$$F_{l,l-1} \approx \mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\mathbf{vec}\left(\mathbf{h}^{(l)} \mathbf{h}^{(l-1)\top} \right) \right] \otimes \mathbb{E}_\pi \left[\mathbf{vec}\left(\delta^{(l)} \delta^{(l-1)\top} \right) \right] \quad (9)$$

として近似できる。$\delta^{(l)}$ および $\delta^{(l-1)}$ はバッチ正規化において操作の対象とならないため $\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\mathbf{vec}\left(\mathbf{h}^{(l)} \mathbf{h}^{(l-1)\top} \right) \right]$ に注目する。$\mathbf{h}^{(l-1)} = (\tilde{W}^{(l)})^{-1}(\tilde{\mathbf{a}}^{(l)} - \tilde{\mathbf{b}}^{(l)})$ に注意すると、これは

$$\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\mathbf{h}^{(l-1)} \mathbf{h}^{(l)\top} \right] = (\tilde{W}^{(l)})^{-1} \overbrace{\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\tilde{\mathbf{a}}^{(l)} f^{(l)}(\tilde{\mathbf{a}}^{(l)})^\top \right]}^{(\dagger)} - (\tilde{W}^{(l)})^{-1} \tilde{\mathbf{b}}^{(l)} \underbrace{\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\mathbf{h}^{(l)} \right]^\top}_{(\ddagger)} \quad (10)$$

と展開できる。

まず式 (10) より、バッチ正規化の導入する位置に関して考察が得られる。まず第二項目が 0 となるためには、(‡) が 0 となれば良い。これは l 層目の線形変換層への入力である $\mathbf{h}^{(l)}$ の期待値であるので、これはバッチ正規化を非線形性の後に導入することにより実現できる。

次に第一項目が 0 となるためには、(†) 部分が 0 となれば良い。もし仮に $f^{(l)}$ が線形であるならば、これを実現するには実用上意味の無い自明な解 $f^{(l)} = 0$ しかあり得ない。しかしながら実用的なニューラルネットワークでは $f^{(l)}$ は非線形であるので、$f^{(l)} \neq 0$ である解が存在する。解の 1 例を示すため、次の確率分布に対する条件を考える。

定義 1. 確率変数 $\mathbf{x} = (x_1, \ldots, x_N)^\top \in \mathbb{R}^N$ の任意の $x_i, x_j \quad (1 \leq i < j \leq N)$ に対して、その周辺化確率密度関数が平均に対して対称、つまり $\xi_i = x_i - \mathbb{E}[x_i]$ および $\xi_j = x_j - \mathbb{E}[x_j]$ に対する確率密度関数 $p(\xi_i, \xi_j)$ が $p(\xi_i, \xi_j) = p(-\xi_i, -\xi_j)$ を満たすとき、$p(\mathbf{x})$ は 2 変数間対称性を持つとする。

この条件を満たす確率変数で想像に易い例としては、対称な確率密度関数、例えば多変量正規分布に従う確率変数が挙げられる。しかしながらこの様な多変量確率密度関数の対称性はきつい条件であり、現実のネットワークを流れる信号が従うと考えるには不自然である。2 変数間対称性は 2 変数間の周辺化確率に対してのみ依存するので、$p(\mathbf{x})$ が非対称の場合にも成り立ちうる、現実により近い緩い条件であると言える。この条件を利用して次のことが言える。

命題 1. $\mathbf{a}^{(l)}$ が 2 変数間対称性を持つとする。また $f^{(l)}$ は偶関数、$f^{(l)}(x) = f^{(l)}(-x)$ だとする。この時、バッチ正規化により $(\dagger) = 0$ となる。

Proof. まず $i = j$ の場合から考える。まず 2 変数間対称性より $\tilde{a}_i^{(l)}$ の周辺確率密度関数 p は偶関数である。また $g(\tilde{a}_i^{(l)}) = \tilde{a}_i^{(l)} f^{(l)}(\tilde{a}_i^{(l)})$ は奇関数である。よって $\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}} \left[\tilde{a}_i^{(l)} f^{(l)}(\tilde{a}_i^{(l)}) \right] = \int g(a) p(a) \mathrm{d}a = 0$ が言える。

次に $i \neq j$ の場合を考える。$^\forall i, j, \quad (1 \leq i < j \leq N)$ に対して $(\tilde{a}_i^{(l)}, \tilde{a}_j^{(l)}) \sim p_{ij}$ とする。また式の読解性のため、

[*2] Kingma and Ba [12] らが論文発表後に考察をしている。

$(a,b) = (\tilde{a}_i^{(l)}, \tilde{a}_j^{(l)})$ と置き換えると、

$$\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}}\left[\tilde{a}_i^{(l)} f^{(l)}(\tilde{a}_j^{(l)})\right] = \iint g(a,b)\, p_{ij}(a,b)\mathrm{d}a\,\mathrm{d}b \qquad (11)$$

とできる。ここで $g(a,b) = af^{(l)}(b)$ とした。$g(-a,-b) = -g(a,b)$ である。これと $p_{ij}(-a,-b) = p_{ij}(a,b)$ から、$\mathbb{E}_{\substack{\mathbf{x}\sim\pi \\ y\sim p}}\left[\tilde{a}_i^{(l)} f^{(l)}(\tilde{a}_j^{(l)})\right] = 0$ である。 □

なお非線形性の後にバッチ正規化を導入した場合においても $f^{(l)}$ の偶奇性に影響しないため、(†) と (‡) は独立に 0 とできることに注意したい。

以上の結論として、クロネッカー因子分解による近似と $\mathbf{a}^{(l)}$ の 2 変数間対称性の条件を認めた場合、非線形性の前後にバッチ正規化を導入することにより偶関数活性化関数を持つネットワークの隣接層パラメータ間に対応するフィッシャー情報量行列の部分を 0 とすることができる。

6. バッチ正規化と他手法との関係性

まず本稿ではクロネッカー因子分解を通してバッチ正規化と自然勾配法の関連を調べた。同様にクロネッカー因子分解を利用した自然勾配近似法 [6, 14] などでは明示的にフィッシャー情報量行列を取り扱っているが、バッチ正規化はネットワーク自体を変化させることによりフィッシャー情報量行列の対角化を実現している点において異なる。クロネッカー因子分解と関係しつつ明示的にフィッシャー情報量行列を扱わない点においては、次に述べる PRONG[5] と似ている。

PRONG[5] は単一層内のパラメータに対応するフィッシャー情報量行列を対角化することを意図して提案されたアルゴリズムであり、アルゴリズムの手続きはバッチ正規化と非常に類似している。本稿の分析から、PRONG が単一層内のパラメータを、そしてバッチ正規化が隣接階層のパラメータに対してそれぞれフィッシャー情報量行列を近似的に対角化しているとの関係性が理解できる。

また本稿で考慮した非線形性の後へのバッチ正規化の導入以外にも、PRONG[5] によっても式 (10) で (‡) = 0 とできる。この意味で PRONG は同時に隣接階層のパラメータ間におけるフィッシャー情報量を低減する効果も持っていると言える。

7. 結論

本稿では数理的な観点からバッチ正規化と自然勾配法との関係を明らかにするため分析を行った。その結果として、クロネッカー因子分解による近似および $\mathbf{a}^{(l)}$ の 2 変数間対称性の条件、そして偶関数活性化関数の利用のもとで、非線形性の前後へのバッチ正規化の導入により隣接層のパラメータ間の学習に関しての自然勾配が実現されうることを示した。

参考文献

[1] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[2] Shun-Ichi Amari, Hyeyoung Park, and Tomoko Ozeki. Singularities affect dynamics of learning in neuromanifolds. *Neural computation*, 18(5):1007–1065, 2006.

[3] KyungHyun Cho, Tapani Raiko, and Alexander T Ihler. Enhanced gradient and adaptive learning rate for training restricted boltzmann machines. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 105–112, 2011.

[4] Florent Cousseau, Tomoko Ozeki, and Shun-ichi Amari. Dynamics of learning in multilayer perceptrons near singularities. *IEEE Transactions on Neural Networks*, 19(8):1313–1328, 2008.

[5] Guillaume Desjardins, Karen Simonyan, Razvan Pascanu, and Koray Kavukcuoglu. Natural neural networks. In *Advances in Neural Information Processing Systems*, pages 2071–2079, 2015.

[6] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. *stat*, 1050:23, 2016.

[7] Roger B Grosse and Ruslan Salakhutdinov. Scaling up natural gradient by sparsely factorizing the inverse fisher matrix. In *ICML*, pages 2304–2313, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] James Martens and Roger B Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*, pages 2408–2417, 2015.

[15] Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

[16] Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2015.

[17] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[18] Nicolas L Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *Advances in neural information processing systems*, pages 849–856, 2008.

[19] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–901, 2016.

[20] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.