G-008

ニューラルネットワークの分散学習における新たな合意重み決定法
New Methods for Setting Consensus Weights in Distributed Training of Neural Networks

原田 和明† 右田 剛史† 高橋 規一†
Kazuaki Harada Tsuyoshi Migita Norikazu Takahashi

1 はじめに

大規模な訓練データをニューラルネットワーク (NN: Neural Network) に学習させるには膨大な計算資源と計算時間が必要である．この問題を解決するため，最近，訓練データを多数の NN に分散的に学習させる方法が提案された [1]．これは，各 NN が，自身に割り当てられた訓練データに対する損失を減少させる処理（局所最適化）と，自身のパラメータ値を近傍のニューラルネットワークのパラメータ値の重み付き平均に置き換える処理（合意更新）を交互に繰り返し行う方法であり，NN 間の通信を表すグラフの構造や合意重み等に関するいくつかの条件の下で収束性が証明されている．本稿では，新たな合意重み決定法を提案し，上記の条件を満たさなくても合意が達成されることや，従来法よりも速く収束することを実験的に示す．

2 ニューラルネットワークの分散学習

Scardapane と Di Lorenzo [1] によって提案された NN の分散学習法について述べる．訓練データ $\mathcal{S}$ を I 個の部分集合 $\mathcal{S}_i = \{(\boldsymbol{x}_{i,m}, d_{i,m})\}_{m=1}^{N_i}$ $(i = 1, 2, \ldots, I)$ に分割し，それらを I 個の NN に割り当てる．ただし $\boldsymbol{x}_{i,m} \in \mathbb{R}^d$, $d_{i,m} \in \mathbb{R}$ である．I 個の NN はすべて同一の構造をもつとする．このとき，i 番目の NN の入力 $\boldsymbol{x} \in \mathbb{R}^d$ に対する出力は $f(\boldsymbol{w}_i; \boldsymbol{x})$ と表される．ここで $\boldsymbol{w}_i \in \mathbb{R}^Q$ は i 番目の NN のすべてのパラメータをまとめたベクトルである．

仮定 1 NN モデル $f(\boldsymbol{w}; \boldsymbol{x})$ は次の二つの条件を満たす．1) f は $\boldsymbol{w}$ に関して連続微分可能である．2) f は $\boldsymbol{w}$ に関してリプシッツ連続である．

NN の分散学習の目標は，各 NN が自身に与えられた訓練データと近傍の NN から受け取るパラメータ値の情報のみを基に自身のパラメータ値を更新していき，全体で次の最適化問題の局所最適解を求めることである．

$$\begin{aligned} &\text{minimize} \quad U(\boldsymbol{w}) = \tfrac{1}{I}\textstyle\sum_{i=1}^{I}\left(\sum_{j=1}^{I} g_j(\boldsymbol{w}_i) + r(\boldsymbol{w}_i)\right) \\ &\text{subject to} \quad \boldsymbol{w}_1 = \boldsymbol{w}_2 = \cdots = \boldsymbol{w}_I \end{aligned} \tag{1}$$

ここで $\boldsymbol{w} = (\boldsymbol{w}_1^{\mathrm{T}}, \boldsymbol{w}_2^{\mathrm{T}}, \ldots, \boldsymbol{w}_I^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{QI}$ はすべての NN のパラメータをまとめたベクトルである．また，$g_j(\boldsymbol{w}_i) = \sum_{m=1}^{N_j} \ell\left(d_{j,m}, f(\boldsymbol{w}_i; \boldsymbol{x}_{j,m})\right)$ である．ℓ は損失関数を表し，$r(\boldsymbol{w}_i)$ は正則化項を表す．i 番目の NN は $\mathcal{S}_i$ 以外の訓練データにはアクセスできないため，$g_j(\boldsymbol{w}_i)$ $(j \neq i)$ を計算できないことに注意する．

仮定 2 最適化問題 (1) 式は次の三つの条件を満たす．1) ℓ は連続微分可能かつリプシッツ連続な凸関数である．2) r は次の (a), (b) のいずれかを満たす．(a) 連続微分可能かつリプシッツ連続な凸関数である．(b) 有界な劣勾配をもつ微分不可能な凸関数である．3) U は強圧的である，すなわち $\lim_{\|\boldsymbol{w}\|\to\infty} U(\boldsymbol{w}) = \infty$ を満たす．

以下では簡単のため，各 NN の通信相手は時間とともに変化しないと仮定する．このとき，NN 間の通信は有向グラフ $G = (\mathcal{V}, \mathcal{E})$ で表される．$\mathcal{V} = \{1, 2, \ldots, I\}$ は I 個の NN に対応する頂点の集合を表し，$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ は有向辺の集合を表す．$\mathcal{E}$ は i 番目の NN が j 番目の NN から情報を受信できるとき，かつそのときに限り (j, i) を含む．ただし $(i, i) \notin \mathcal{E}$ とする．頂点 $i \in \mathcal{V}$ の入近傍と出近傍を，それぞれ $\mathcal{N}_i^{\mathrm{in}} = \{j \mid (j, i) \in \mathcal{E}\} \cup \{i\}$, $\mathcal{N}_i^{\mathrm{out}} = \{j \mid (i, j) \in \mathcal{E}\} \cup \{i\}$ で定義する．

Scardapane と Di Lorenzo [1] の分散学習アルゴリズムでは，各 NN は局所最適化と合意更新を交互に行う．n 回反復後の i 番目の NN のパラメータ値を $\boldsymbol{w}_i[n]$ とし，$\sum_{j=1, j\neq i}^{I} \nabla_{\boldsymbol{w}_i} g_j(\boldsymbol{w}_i[n])$ の推定値を $\tilde{\boldsymbol{\pi}}_i[n]$ とする．各 NN は，局所最適化として $\boldsymbol{w}_i$ に関する次の最適化問題を解く．

$$\text{minimize} \quad \tilde{g}_i(\boldsymbol{w}_i; \boldsymbol{w}_i[n]) + \tilde{\boldsymbol{\pi}}_i[n]^{\mathrm{T}}(\boldsymbol{w}_i - \boldsymbol{w}_i[n]) + r(\boldsymbol{w}_i) \tag{2}$$

ここで $\tilde{g}_i(\boldsymbol{w}_i; \boldsymbol{w}_i[n])$ は，$g_i(\boldsymbol{w}_i)$ の $\boldsymbol{w}_i = \boldsymbol{w}_i[n]$ における強凸代理関数である．本稿では，g_i の $\boldsymbol{w}_i = \boldsymbol{w}_i[n]$ における線形近似関数に近接正則化項を加えた $\tilde{g}_i(\boldsymbol{w}_i; \boldsymbol{w}_i[n]) = g_i(\boldsymbol{w}_i[n]) + \nabla g_i(\boldsymbol{w}_i[n])^{\mathrm{T}}(\boldsymbol{w}_i - \boldsymbol{w}_i[n]) + \frac{\tau}{2}\|\boldsymbol{w}_i - \boldsymbol{w}_i[n]\|_2^2$ を代理関数に用いる．ただし τ は正定数である．各 NN は (2) の最適解 $\tilde{\boldsymbol{w}}[n]$ を求め，$\boldsymbol{z}_i[n] = \boldsymbol{w}_i[n] + \alpha[n](\tilde{\boldsymbol{w}}_i[n] - \boldsymbol{w}_i[n])$ を求める．ここで $\{\alpha[n]\}_{n=0}^{\infty}$ は，$\alpha[0], \epsilon \in (0, 1]$, $\alpha[n] = \alpha[n-1](1 - \epsilon\alpha[n-1])$ $(n = 1, 2, \ldots)$ によって決まる数列である．合意更新では，各 NN は $\boldsymbol{w}_i[n+1]$, $\boldsymbol{y}_i[n]$ と $\tilde{\boldsymbol{\pi}}_i[n+1]$ をそれぞれ $\boldsymbol{w}_i[n+1] = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}} c_{ij}\boldsymbol{z}_j[n]$, $\boldsymbol{y}_i[n+1] = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}} c_{ij}\boldsymbol{y}_j[n] + \nabla g_i(\boldsymbol{w}_i[n+1]) - \nabla g_i(\boldsymbol{w}_i[n])$, $\tilde{\boldsymbol{\pi}}_i[n+1] = I\boldsymbol{y}_i[n+1] - \nabla g_i[n+1]$ によって計算する．ここで c_{ij} は合意重みとよばれる非負定数である．また，$\boldsymbol{y}_i[n]$ と $\tilde{\boldsymbol{\pi}}_i[n]$ はそれぞれ $\frac{1}{I}\sum_{j=1}^{I} \nabla_{\boldsymbol{w}_i} g_j(\boldsymbol{w}_i[n])$ と $\sum_{j=1, j\neq i}^{I} \nabla_{\boldsymbol{w}_i} g_j(\boldsymbol{w}_i[n])$ の推定値を表す．詳細は [1] を参照されたい．

以上の分散学習アルゴリズムに対して次が成り立つ．

命題 1（文献 [1] の Proposition 2 の簡潔版） 次の四つの仮定が成り立つとき，分散学習アルゴリズムによって生成されるすべての点列の極限 $\{(\boldsymbol{w}_1[n]^{\mathrm{T}}, \boldsymbol{w}_2[n]^{\mathrm{T}}, \ldots, \boldsymbol{w}_I[n]^{\mathrm{T}})^{\mathrm{T}}\}_{n=0}^{\infty}$ は (1) の停留点である．1) 仮定 1, 2 が成り立つ．2) $G = (\mathcal{V}, \mathcal{E})$ は強連結である．3) 合意重み c_{ij} は正定数で

$$\sum_{j\in\mathcal{N}_i^{\mathrm{in}}} c_{ij} = 1, \quad \forall i \in \{1, 2, \ldots, I\}, \tag{3}$$

$$\sum_{j\in\mathcal{N}_i^{\mathrm{out}}} c_{ji} = 1, \quad \forall i \in \{1, 2, \ldots, I\}, \tag{4}$$

を満たす．4) 任意の n に対して $\alpha[n] \in (0, 1]$ であり，かつ $\sum_{n=0}^{\infty} \alpha[n] = \infty$ が成り立つ．

† 岡山大学大学院自然科学研究科 Graduate School of Natural Science and Technology, Okayama University

3　合意重み決定法

以下では，簡単のため NN のつながりは対称とする．すなわち，$(j,i) \in \mathcal{E}$ であるとき，かつそのときに限り $(i,j) \in \mathcal{E}$ とする．NN の分散学習において，合意重みをどのように決めるかは収束性に関わる重要な問題であり，これまでにいくつかの方法が提案されている．

Blondel ら [2] の合意アルゴリズムでは，次式の合意重み決定法が用いられている．

$$c_{ij} = \begin{cases} 1/(\delta_i + s), & \text{if } j \in \mathcal{N}_i^{\text{in}} \setminus \{i\} \\ s/(\delta_i + s), & \text{if } j = i \end{cases} \tag{5}$$

ここで δ_i はグラフ G の頂点 i の次数であり，$s \in \{0,1\}$ である．式 (5) は $s = 0$ と $s = 1$ の両方の場合に (3) を満たすが，一般に (4) を満たさない．

Scardapane と Di Lorenzo [1] の実験では，次式で表される Metropolis-Hasting (MH) 法が用いられている．

$$c_{ij} = \begin{cases} 1/(\max\{\delta_i, \delta_j\} + 1), & \text{if } j \in \mathcal{N}_i^{\text{in}} \setminus \{i\} \\ 1 - \sum_{j \in \mathcal{N}_i^{\text{in}}} 1/(\max\{\delta_i, \delta_j\} + 1), & \text{if } j = i \end{cases} \tag{6}$$

式 (6) によって決まる合意重みは $c_{ij} = c_{ji}$ を満たすので，(3) と (4) が同時に満たされる．

Sun ら [3] は NN の収束性の証明に (3) は不要である[1]と主張して次式の重み決定法を提案した．

$$c_{ij} = \begin{cases} 1/(\delta_j + s), & \text{if } j \in \mathcal{N}_i^{\text{in}} \setminus \{i\} \\ s/(\delta_j + s), & \text{if } j = i \end{cases} \tag{7}$$

ここで $s \in \{0,1\}$[2]である．式 (7) によって決まる重みは $s = 0$ と $s = 1$ の両方の場合に (4) を満たす．

本稿では次式の合意重み決定法を提案する．

$$c_{ij} = \begin{cases} \delta_j/(\sum_{j \in \mathcal{N}_i^{\text{in}} \setminus \{i\}} \delta_j + s), & \text{if } j \in \mathcal{N}_i^{\text{in}} \setminus \{i\} \\ s/(\sum_{j \in \mathcal{N}_i^{\text{in}} \setminus \{i\}} \delta_j + s), & \text{if } j = i \end{cases} \tag{8}$$

ここで $s \in \{0,1\}$ である．式 (5) と同じく，(8) は (3) を満たすが，一般に (4) を満たさない．提案手法の重要な特徴は c_{ij} $(j \in \mathcal{N}_i^{\text{in}} \setminus \{i\})$ の値が頂点 j の次数に比例することである．多くの NN とつながっている NN ほど多くの情報を持っていると考えられるため，次数の高い NN の重みを大きく設定することで NN 間の合意が速く達成されると期待される．

4　数値実験

合意重みの決定法と NN の分散学習の収束性の関係を明らかにするため Wisconsin (Breast Cancer Wisconsin) データセットを用いて実験を行った．このデータセットの入力ベクトルの次元は $d = 9$ であり，データの総数は 689 である．これを互いに素な 10 個の部分集合に分け，図 1 のようにつながっている 10 個の NN に割り当てた．各 NN は入力層，中間層，出力層の 3 層からなる．中間層のニューロン数を 10 とし，すべてのニューロンの活性化関数にはハイパボリックタンジェントを用いた．損失関数と正則化項は文献 [1] と同じく $\ell(a,b) = -a\log(b) - (1-a)\log(1-b)$, $r(\boldsymbol{w}) = \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$

1)　いくつかの合意アルゴリズムでは仮定 (4) のみを用いて収束性が証明されている [4,5]

2)　厳密には，彼らは (7) の $s = 0$ 場合しか提案していない．

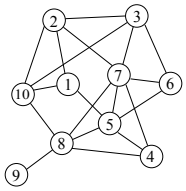

図 1　NN の通信を表すグラフ

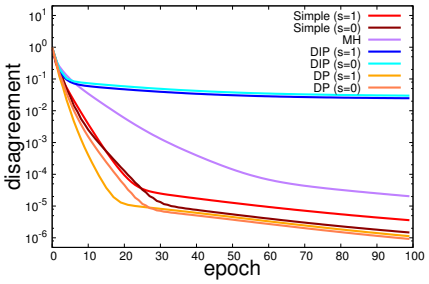

図 2　合意誤差の値の時間推移

とした．NN のパラメータは標準正規分布に従う乱数で初期化する．ハイパーパラメータは $\tau = 1.0$, $\lambda = 15$, $\alpha[0] = 0.03$, $\epsilon = 0.01$ とする（これらの値は予備実験によって決定した）．すべてのアルゴリズムを Python 3.6.6 で実装し，Intel Core i5-4590 と 8GB RAM のメモリを搭載する PC 上で実行した．

合意誤差 $D[n] = \frac{1}{10}\sum_{i=1}^{10}\left\|\boldsymbol{w}_i[n] - \frac{1}{10}\sum_{j=1}^{10}\boldsymbol{w}_j[n]\right\|_2$ の時間変化を図 2 に示す．Simple, MH, DIP (Degree Inversely Proportional), DP (Degree Proportional) はそれぞれ (5), (6), (7), (8) を表す．合意誤差が最も速く減少したのは DP であった．2 番目が Simple であり，その他の方法は大幅に遅かった．また，DP ($s = 0$) が DP ($s = 1$) より速く収束しており，Simple でも同様と結果となった．これは，各 NN が重み付き平均を計算する際に自身の値を除外する方がよいことを示している．目的関数 (1) については，どの方法を用いても同じように減少しており，7 種類の方法に大きな違いは見られなかった．

5　まとめ

ニューラルネットワークの分散学習における新たな合意重み決定法を提案し，従来法よりも速く合意形成を達成できることを実験的に示した．今後の課題は，多様なデータセットを用いた検証と収束性の理論解析である．

参考文献

[1] S. Scardapane and P. Di Lorenzo, “A framework for parallel and distributed training of neural networks,” *Neural Networks*, vol.91, pp.42–54 (2017).

[2] V. D. Blondel, J. M. Hendrickx, A. Olshevsky and J. N. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” *Proceedings of the 44th IEEE Conference on Decision and Control*, pp.2996–3000 (2005).

[3] Y. Sun, G. Scutari and D. Palomar, “Distributed nonconvex multiagent optimization over time-varying networks,” *Proceedings of the 50th Annual Asilomar Conference on Signals, Systems, and Computers* (2016).

[4] A. Nedić and J. Liu, “On convergence rate of weighted-averaging dynamics for consensus problems,” *IEEE Transactions on Automatic Control*, vol.62, no.2, pp.766–781 (2017).

[5] N. Takahashi and K. Kawashima, “A simple sufficient condition for convergence of projected consensus algorithm,” *IEEE Control Systems Letters*, vol.2, no.3, pp.537–542 (2018).