

## Cost Sensitive 学習に対する重み付きバッチ正規化と重み付き入力正規化

Yeo Xian En<sup>†</sup>山形大学大学院理工学研究科<sup>†</sup>安田宗樹<sup>‡</sup>山形大学大学院理工学研究科<sup>‡</sup>

## 1 はじめに

Cost Sensitive (CS) 学習は機械学習 [1, 2] の中でとても重要な学習法であり、データ毎に異なる重みつけをもった損失関数を最適化する学習である。CS 学習はデータに重みを付けるので不均衡データでの学習や、共変量シフト学習などに学習の改善に応用できるが、DNN 上での CS 学習において通常のバッチ正規化 [3] と一緒に組み合わせて使うと相性が悪いという問題が発生してしまう。そこで、データ点の重みを考慮して改良した重み付きバッチ正規化 [4] が問題を解決できることが報告されている。本研究では、重み付きバッチ正規化に加え、CS 学習を高性能化するために入力データ前処理においてもデータ点の重みを考慮する正規化法を提案する。

## 2 Cost Sensitive 学習

$N$  個のデータ点で構成される訓練データ集合  $D := \{(\mathbf{x}_\mu, \mathbf{t}_\mu) \mid \mu = 1, 2, \dots, N\}$  があるとす。ここで、入力データ  $\mathbf{x}$  と対応する教師データ  $\mathbf{t}$  の組を一つのデータ点  $(\mathbf{x}, \mathbf{t})$  として、数が  $\mu$  個である。この訓練データ集合に対して、DNN での損失関数は

$$L(\theta) := \frac{1}{N} \sum_{\mu=1}^N f(\mathbf{x}_\mu, \mathbf{t}_\mu, \theta) \quad (1)$$

である。ここで、 $\theta$  は学習パラメータであり、 $\ell(\mathbf{x}_\mu, \mathbf{t}_\mu, \theta)$  は  $\mu$  番目の訓練データに対する損失を表わしている。

CS 学習は次のような重み付き損失関数

$$L_w(\theta) := \frac{1}{Z} \sum_{\mu=1}^N w_\mu f(\mathbf{x}_\mu, \mathbf{t}_\mu, \theta) \quad (2)$$

と定式化される。 $w_\mu$  は  $\mu$  番目の訓練データに対する正の重み係数であり、 $Z := \sum_{\mu=1}^N w_\mu$  は規格化定数である。重み係数  $w_\mu$  は対応するデータ点の重みを変えることによって、データの実質的なデータ数を調整されていると考えられる。例えば、 $w_\mu = 100$  の場合、 $\mu$  番目の訓練データが 100 個に複製されていると考えられる。

こうして、重み係数に適切な値に設定すれば、データが不均衡の環境でも学習が可能になる。重み係数の設定法は様々であるが、クラス分類問題に対しては inverse class frequency (ICF)[5] が多く用いられている、一方、不均衡回帰問題の場合

では inverse kernel density estimation (IKDE) [6] が有効であることが確認されている。ICF は、重み係数が該当するデータの確率の逆数：

$$w_\mu = \left(\frac{n(\mu)}{N}\right)^{-1} \quad (3)$$

と設定する。 $n(\mu)$  は  $\mu$  番目の訓練データが属するクラスに属するデータ数のこと。ICF はクラスごとのデータ数をバランスを調整し、疑似的に均衡データの学習を実現することができる。

一方、回帰問題の場合では学習するデータは連続実数値のためクラス分類問題のようにデータがクラスに属してないので ICF を用いることがとても困難である。そこで、

$$P(\mathbf{t}) = \frac{1}{Nh^m} \sum_{\mu=1}^N K\left(\frac{\mathbf{t} - \mathbf{t}_\mu}{h}\right) \quad (4)$$

のように KDE [7] を用いてデータの確率密度分布を計算する。ここで、 $K(\mathbf{t})$  はカーネル関数であり、 $h$  はバンド幅 (分布の平滑性) を表わすハイパパラメータである。また、 $m$  は出力次元である。そして ICF の概念を応用して、 $\mu$  番目の訓練データに対する重みを次のように分布  $P(\mathbf{t})$  の逆数 (inverse KDE (IKDE)) で設定する。

$$w_\mu = P(\mathbf{t}_\mu)^{-1} \quad (5)$$

## 3 重み付きバッチ正規化

バッチ正規化は、中間層の出力信号をミニバッチ毎に標準化し、平均・分散を一定に保つようする [3]。中間層の出力信号の分布が中間層を経るごとに大きく変化してしまうこと (共変量シフト) を防ぎ、学習の質を向上されることができる。

しかしながら、バッチ正規化における標準化は CS 学習と相性が悪い。CS 学習の重み係数  $w_\mu$  によってデータの実質的なサイズは変化する一方、通常のバッチ正規化は重み係数  $w_\mu$  によるデータサイズの変化を考慮しないので、結果としてデータサイズの解釈に不一致が生じてしまうのである。そこで、不一致の問題を解消する方法として WBN が提案された [4]。WBN を使用することによって、CS 学習の重み係数  $w_\mu$  を考慮した標準化計算が可能となり、データサイズの解釈を CS 学習とバッチ正規化の両方において一致させることができる。

## 4 重み付き入力標準化

入力標準化は学習する前に入力データを平均 0、分散 1 に標準化する作業のことであり、z-score 法と呼ばれる：

$$\hat{x}^{(\mu)} = \frac{x^{(\mu)} - x_{\text{mean}}}{x_{\text{std}}} \quad (6)$$

Weighted Batch Normalization and Weighted Standardization for Cost - Sensitive Learning

<sup>†</sup> Yeo Xian En; Graduate School of Science and Engineering, Yamagata University

<sup>‡</sup> Muneki Yasuda; Graduate School of Science and Engineering, Yamagata University

ここで、 $x_{\text{mean}}$  はデータの平均、 $x_{\text{std}}$  はデータの標準偏差である。しかしながら、z-score 法もバッチ正規化法と同じように CS 学習において相性が悪いと考えられる。CS 学習は重み係数  $w_{\mu}$  によってデータの実質的なデータサイズを変更するが、通常の入力の標準化はそれを考慮していない。本研究は重み付きバッチ正規化と共に、CS 学習の性能をさらに良くするために重み付き入力標準化 (weighted z-score (WZS)) を提案する。提案法では、データの平均：

$$x_{\text{mean}} = \frac{1}{Z} \sum_{\mu=1}^N w_{\mu} x^{(\mu)} \quad (7)$$

と (不偏) 分散：

$$x_{\text{std}}^2 = \frac{Z}{Z^2 - \sum_{\mu=1}^N w_{\mu}^2} \sum_{\mu=1}^N w_{\mu} (x^{(\mu)} - x_{\text{mean}})^2 \quad (8)$$

を用いて、式 (6) の方法により標準化を行う。このように、データの実質的なデータ数を考慮しながら標準化する。

#### 4.1 数値実験

本研究は MNIST[8] データセットを使って重み付き入力正規化を検証する。MNIST は 28x28 ピクセルの手書き数字画像データのことであり、CS 学習を有効活用するため MNIST データセットからクラス 3, 4, 7 と 9 を用いて、さらに不均衡環境を作るためにクラス 3, 4, 7 の訓練データ数を 5 と設定する。テストデータはそれぞれのクラスにあるすべてのテストデータを使う。DNN のモデルは 784 素子数の入力層、300 素子数の中間層、300 素子数の中間層と 4 素子数の出力層から構成されている。中間層の活性化関数は ReLU、出力層の活性化関数は恒等関数を使用している。損失関数はクロスエントロピーとし、最適化法はミニバッチサイズ 128 の adamax を利用した。

表 1 MNIST 実験のデータ数

MNIST データ	クラス 3	クラス 4	クラス 7	クラス 9
訓練データ	5	5	5	5000
テストデータ	1010	982	1028	1009

表 2 に 4 種類の実験モデルそれぞれのクラス認識率と全体認識率を表す。それは普通の損失関数とバッチ正規化、CS 学習とバッチ正規化の単純に組み合わせ、CS 学習と重み付きバッチ正規化、CS 学習と重み付きバッチ正規化と重み付き入力正規化の 4 種類である。

表 2 200 エポック学習後の DNN でテストデータを評価したときの損失値。10 回の実験の平均値である。認識率は % 表記である。

モデル	クラス 3	クラス 4	クラス 7	クラス 9	全体
BN	13.7	3.4	6.0	100.0	30.8
CS(ICF)+BN	15.3	4.8	9.9	100.0	32.6
CS(ICF)+WBN	57.4	40.3	42.9	99.5	60.1
CS(ICF)+WBN+WZS (提案法)	70.3	37.2	48.4	100.0	64.1

表 2 の実験結果から不均衡問題の学習のために CS 学習とバッチ正規化を用いたが結果はそれほど改善しないというこ

とが分かる。その理由はデータサイズへの解釈がやはり違う。そこで、重み付きバッチ正規化を使うとクラス認識率が全体的に改善される。そして、今回本研究で CS 学習をさらに改善したために WZS を提案した提案法は全体的な認識率を 64.1% まで向上した。この結果からデータの入力正規化が非常に重要であることが分かって、CS 学習を行う際に重み付き入力正規化を適用した方がいいことが分かる。

## 5 まとめ

本研究では CS 学習能力を向上させるために重み付きバッチ正規化法のほかに、重み付き入力正規化法を提案した。そして、不均衡 MNIST データの実験結果から重み付き入力正規化法は CS 学習に有効であることが確認できた。しかし、本研究では不均衡データのみに対する CS 学習の結果しか検証していないため、今後の課題として CS 学習が適用できる問題、例えば共変量シフト問題に適用し、提案法の信頼性を高めたいと考えられている。

## 謝辞

本研究は科研費 (18K11459, 18H03303, 21K11778) 及び JST CREST (JPMJCR1402) の助成を受けたものである。

## 文献

- [1] 麻生ら: 深層学習, 近代科学社, 2015.
- [2] I. Goodfellow, Y. Bengio and A. Courville: Deep Learning, MIT Press, 2016
- [3] S. Ioffe and C. Szegedy: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *In Proc. of the 32nd International Conference on International Conference on Machine Learning (ICML)*, vol.37, pp.448-456, 2015
- [4] M. Yasuda, Y. X. En, and S. Ueno: Consistent Batch Normalization for Weighted Loss in Imbalanced-Data Environment, *Nonlinear Theory and its Applications (NOLTA), IEICE*, Vol.11, Issue 4, pp.454-465, 2020.
- [5] C. Huang, Y. Li, C. C. Loy, and X. Tang: Learning Deep Representation for Imbalanced Classification, *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.5375-5384, 2016
- [6] 楊顯恩, 安田宗樹: 不均衡回帰問題に対する重み付きバッチ正規化法の検証, 情報処理学会 第 83 回全国大会, 2021
- [7] C. M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006
- [8] Y. LeCun, L. Bottou, Y. Bengio and P.Haffner: Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11), pp.2278-2324, November 1998