

Outlier-Robust Neural Network Training: Efficient Optimization of Transformed Trimmed Loss with Variation Regularization (Submitted.)

奥野彰文^{1,2}, 柳下翔太郎¹

¹統計数理研究所, ²理研AIP

Outlier-Robust Neural Network Training

なにをしたのか

▶ 外れ値にロバストかつフレキシブルな予測法の提案

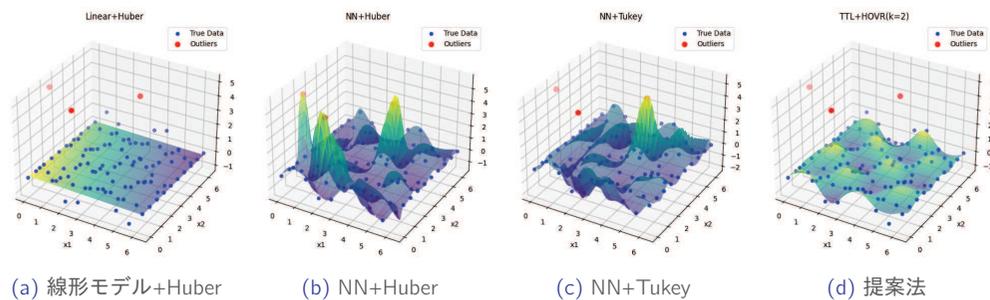
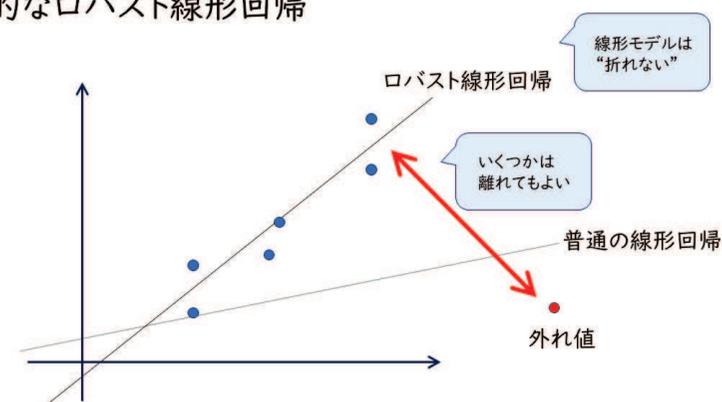
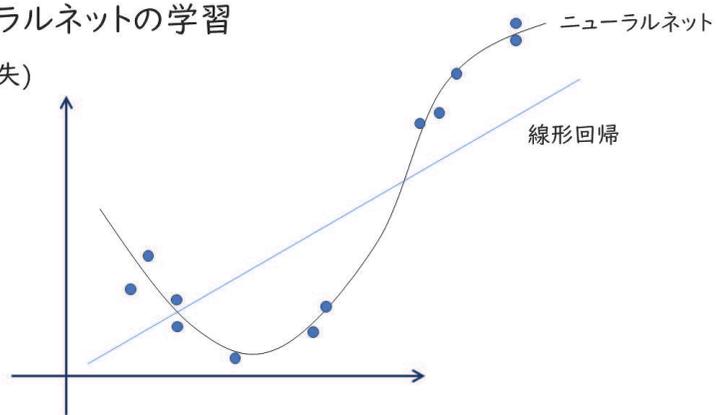


Figure: $f_*(x) = \sin(2x_1) \cos(2x_2)$.

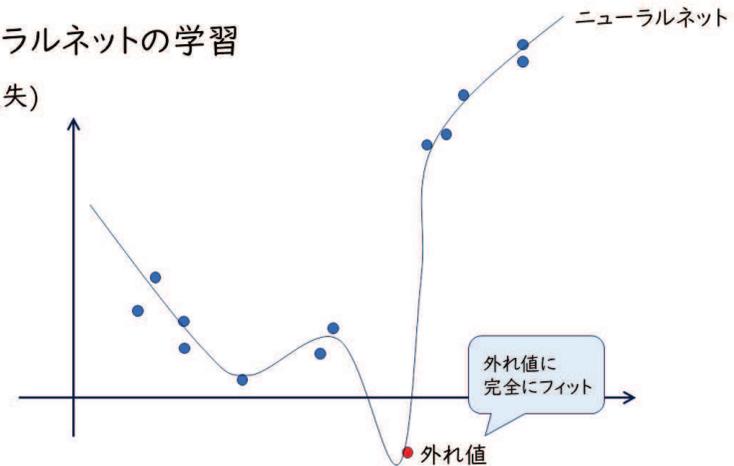
典型的なロバスト線形回帰



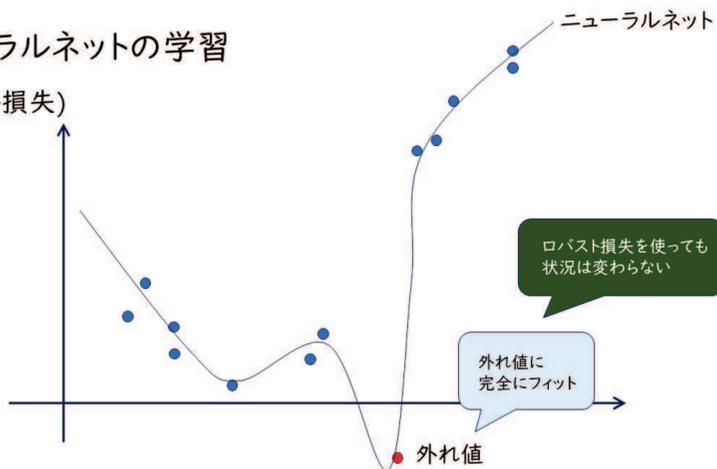
ニューラルネットの学習
(二乗損失)



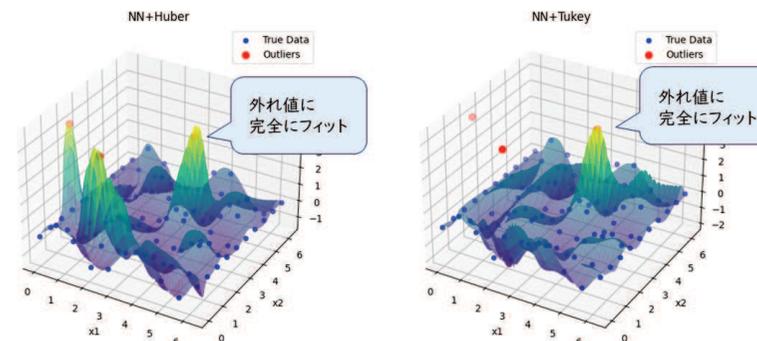
ニューラルネットの学習
(二乗損失)



ニューラルネットの学習
(ロバスト損失)

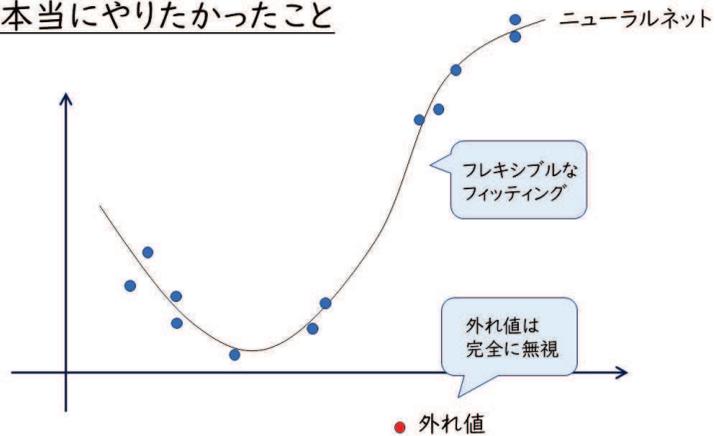


実際のロバスト損失+NN (+勾配法)



つまり何が言いたいのかというと

我々が本当にやりたかったこと



- ▶ 予測関数 $f_{\theta}(x)$ はいい感じにグニャグニャしてほしいが、
- ▶ 過剰にグニャグニャしないほしい。

予測関数に“硬さ”のようなものを入れたい。

関数の変動を抑える

Higher-Order Variation Regularization (HOVR; Okuno, arXiv:2308.02293v1):

$$C_{k,q}(f_{\theta}) := \int_{\Omega} \left| \frac{\partial^k f_{\theta}(x)}{\partial x^k} \right|^q dx.$$

例えば正規直交基底を用いた回帰関数 $f_{\theta}(x) = \sum_j \theta_j \phi_j(x)$ の場合には、

$$C_{0,2}(f_{\theta}) \propto \|\theta\|_2^2$$

なので、パラメータ正則化の一般化と思える。

- ▶ 導関数は autograd で計算可能。
- ▶ 積分は(明示的に計算せず)確率的に回避できる。

つまりどうこと?

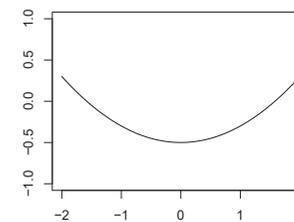


Figure: $C_{2,2}(f) \approx 0.64$

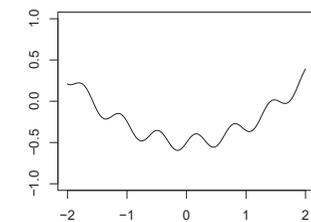


Figure: $C_{2,2}(f) \approx 197$

- ▶ 激しいグニャグニャを抑制できる。
- ▶ 線形/カーネル回帰モデルなどではパラメータ正則化に対応。
- ▶ ニューラルネットワークでは???

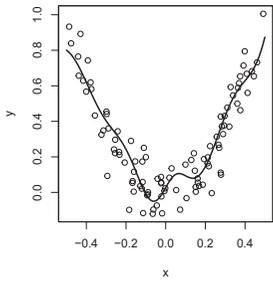


Figure: Weight decay (要するにリッジ)

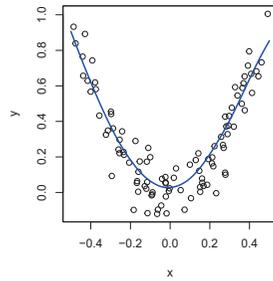
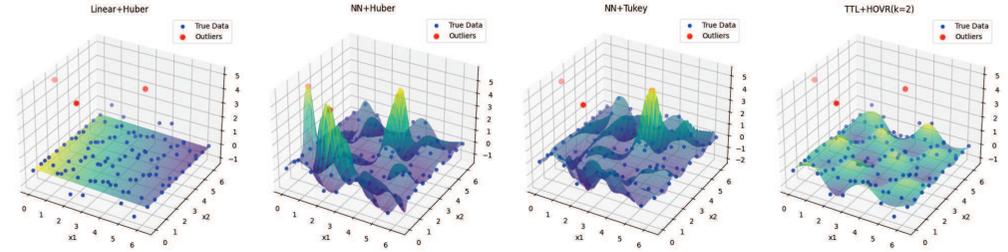


Figure: HOVR

- ▶ Akifumi Okuno, "A stochastic optimization approach to train non-linear neural networks with regularization of higher-order total variation", arXiv:2308.02293v1.
- ▶ デスクリジェクトされて萎えた。放置 ⇒ 気付いたら1年が経過していました…

というわけで最初の図に戻る。

- ▶ (入力)2次元 - 100 - 100 - 100 - (出力)1次元の多層パーセプトロン。
- ▶ パラメータ数2万超, 外れ値3%混入。



(a) 線形モデル+Huber

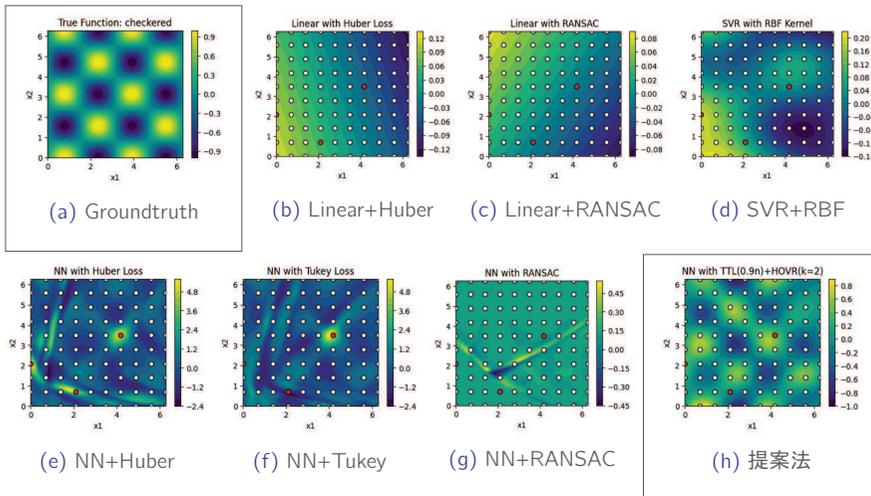
(b) NN+Huber

(c) NN+Tukey

(d) 提案法

Figure: $f_*(x) = \sin(2x_1) \cos(2x_2)$.

やってみた 1 (checked)



まとめ

- ▶ ロバスト損失を使っても、ニューラルネットは外れ値に過適合してしまう。
- ▶ ニューラルネットを“硬くする”正則化としてHOVRを提案。
- ▶ ロバスト損失+正則化と等価なARTLを提案。
- ▶ ARTLを最小化する確率的アルゴリズムSGSDを提案。
- ▶ 収束証明もできたし、その実験結果もよさそうだった。

コメント・ご質問は okuno@ism.ac.jp まで。