Scientific Research Publishing

Open Journal of Statistics, 2026, 16(1), 1-15
https://www.scirp.org/journal/ojs
ISSN Online: 2161-7198
ISSN Print: 2161-718X

Estimating the Rectified Linear Unit Activation Function in Deep Learning Using the Grouping-Adjusted Median Estimator: An Evaluation of Max Pooling

Kazumitsu Nawata

University of Tokyo (Retired), Tokyo, Japan
Email: nawatak@g.ecc.u-tokyo.ac.jp, kn1016abc@gmail.com

How to cite this paper: Nawata, K. (2026) Estimating the Rectified Linear Unit Activation Function in Deep Learning Using the Grouping-Adjusted Median Estimator: An Evaluation of Max Pooling. *Open Journal of Statistics*, **16**, 1-15.
https://doi.org/10.4236/ojs.2026.161001

Received: December 7, 2025
Accepted: January 17, 2026
Published: January 20, 2026

Abstract

The Rectified Linear Unit (ReLU) activation function is widely employed in deep learning (DL). ReLU shares structural similarities with censored regression and Tobit models common in econometrics and statistics. Although the conventional Tobit maximum likelihood estimator (CTMLE) is frequently applied in these fields, the insights gained have not been fully incorporated into DL research. When models are affected by random observation errors or distributional misspecification, CTMLE often exhibits substantial bias. To address this limitation, we consider the Grouping Adjusted Median Estimator (GAME). GAME is a robust method that does not rely on specific distributional assumptions and is constructed through three stages: grouping, adjustment, and computation of adjusted medians combined with weighted Tobit maximum likelihood estimation. Monte Carlo experiments show that GAME outperforms CTMLE in non-standard settings while incurring only minor efficiency losses under standard conditions. Max pooling is widely used in DL, and its mechanism resembles grouping. This study also evaluates the max pooling estimator (MPE). However, MPE often performs poorly, as the estimated boundaries between target and background regions may be reversed relative to the true ones. While max pooling can be useful for detecting weak signals, it may yield misleading results in high-noise situations, necessitating special care. GAME offers a promising alternative for mitigating noise effects, and the median can be generalized to arbitrary percentiles, providing additional flexibility in estimation.

Keywords

Deep Learning, ReLU, Max Pooling, Tobit Model, Median, Grouping,

Adjustment, Robust Estimation

1. Introduction

The Rectified Linear Unit activation function (ReLU) [1]-[11], along with its modifications such as leaky ReLU [12]-[15], is widely used in deep learning (DL), especially in two-dimensional (2D) and three-dimensional (3D) image analysis. We focus on input layers in 2D or 3D image analysis, but our arguments easily extend to intermediate layers and multi-dimensional problems. ReLU is given by $h(a)=\max(0,a)$ and can be expressed as $y=h(x'w+u)$ when random noise is present. For image analysis, x and w represent vectors of pixel values and their corresponding weights, respectively. Analyzing data with ReLU is similar to censored regression or Tobit models [16] (pp. 973-977) that are widely studied in statistics and econometrics [17]-[19]. Tobit was introduced by Tobin [20] in his analysis of household expenditures on durable goods. However, knowledge from these studies has not been fully incorporated into DL research. The results derived from econometrics and statistics resemble those of DL, and incorporating these applications may enhance the performance of DL methods.

Pooling techniques [21]-[26], especially max pooling [27]-[32], have also been widely used. However, max pooling suffers from a serious drawback: when noise is large and signal is small, it often produces incorrect results. Song *et al.* [33] consider median-pooled gradients and note that "median pooling is suitable to reduce the effect of noise" [33] (p. 140). Nevertheless, their study lacks a solid theoretical framework, and median pooling itself is rarely employed.

Nawata [34] proposes the Grouping Adjusted Median Estimator (GAME), which is robust to non-normality and heteroscedasticity in the Tobit model. Grouping is conceptually similar to pooling, and the asymptotic distribution of GAME has been derived. In this paper, we examine the application of GAME to the Tobit model and present results from Monte Carlo experiments. When a signal represents a small percentage of the area, we can use an appropriate percentile. The concept of the "median" can be generalized to arbitrary percentiles.

2. Models and Game

2.1. Models

In image analysis, it is important to estimate the boundary between two regions, such as target and background. For example, Figure 9.2 in Bishop and Bishop [35] (p. 260) shows contours of cat images in which the cat's position differs across images. When the positions are corrected, the results become identical.

We consider cases in which the boundary is given by a linear function $x_i'w=0$, where x_i are vectors of explanatory variables including a constant term and w are their coefficients (weights), respectively. Since multidimensional splines may be used [36] [37], the boundary can be approximated by a linear function, at least

locally.

When noise or an error term exists, the observed dependent variable y_i is given by

$$y_i^* = x_i'w + u_i, \quad y_i = \max\left(0, y_i^*\right), \quad i = 1, 2, \cdots, n, \tag{1}$$

where u_i is an error term and n is the total number of observations (pixels). For mono-colored grayscale images drawn on white paper with a brush, y_i represents the darkness at each point (pixel) and x_i represents the location (including its functional transformations) of the point. If the censoring threshold is not zero, use $\min\{y_1, y_2, \cdots, y_n\}$ [17].

In statistics and econometrics, the unknown parameters in Equation (1) are usually estimated by the conventional Tobit maximum likelihood estimator (CTMLE), which assumes normal error terms and maximizes the likelihood function:

$$L\left(w, \sigma^2\right) = \prod_{y_i>0} \frac{1}{\sigma}\phi\left[\left(y_i - x_i'w\right)/\sigma\right] \prod_{y_i=0} \Phi\left[-\left(y_j - x_i'w\right)/\sigma\right], \tag{2}$$

where ϕ and Φ are the density and distribution functions of the standard normal distribution. However, CTMLE is not consistent and often exhibits large biases when the error terms are non-normal or heteroscedastic [38] (pp. 378-381).

2.2. Grouping Game

Nawata [34] proposes GAME for the Tobit model. The estimation method consists of the following three stages, and **Figure 1** is a flowchart of the estimation procedure.

i) Grouping Stage

We assume that the sample space of x_i, denoted by S, is bounded (if not, consider a proper bounded subset of S). We divide S into m non-overlapping groups or windows, $\{S_j\} = \{S_1, S_2, \cdots, S_m\}$, satisfying the following conditions. Pooling typically uses square or rectangular windows, whereas grouping can take on more freeform shapes.

Condition 1. The distance between any two points in each group converges to zero as n goes to infinity.

Condition 2. Let n_j and n_j^+ be the numbers of observations in I_j and I_j^+, respectively, where I_j^+, I_j^0, and I_j are the index sets given by $I_j^+ = \{i : x_i \in S_j \text{ and } y_i > 0\}$, $I_j^0 = \{i : x_i \in S_j \text{ and } y_i = 0\}$, and $I_j = \{i : x_i \in S_j\} = I_j^+ \cup I_j^0$. $\{n_j\} = \{n_1, n_2, \cdots, n_m\}$ increase on the order of n^δ and $\frac{n_j^+}{n_j} > a$ for some $0 < \delta \le 1$ and $a > 1/2$. There must be enough groups for the model to be estimable (otherwise, change the grouping or replace the median with an appropriate percentile).

In image analysis, pixels are divided into grid windows to form groups. Note that if x_i is discrete, the data are grouped by the values taken by x_i, and the group sizes become zero. Hereafter, we denote (x_i, y_i) for $i \in I_j$ as (x_{ij}, y_{ij}). Let $\bar{x}_j$ de-

note the average of x_{ij} $(=\sum_i x_{ij}/n_j)$, let $\bar{y}_j$ denote the median of $\{y_{ij}\}=\{y_i : i \in I_j\}$, and let Λ^+ denote the index set defined by $\Lambda^+ = \left\{ j : \frac{n_j^+}{n_j} > a \right\}$. To avoid unnecessary complications, the median is defined as the $(n_j+1)/2$-th largest value for odd values and $(\frac{n_j}{2}+1)$-th largest value for even values. Since $\bar{y}_j$ is the median, it takes positive values for $j \in \Lambda^+$. The model and its ordinary least squares (OLS) estimator for $\{(\bar{x}_j, \bar{y}_j) : j \in \Lambda^+\}$ are given by:

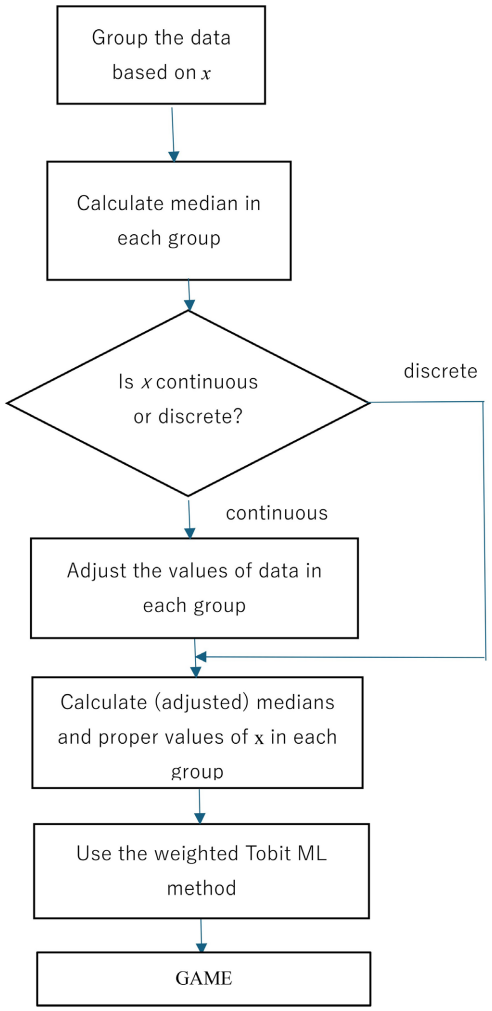

Figure 1. Flowchart of the Grouping Adjusted Estimator (GAME).

$$\bar{y}_j = \bar{x}_j' w + e_j, \quad j \in \Lambda^+, \tag{3}$$

$$\hat{w}_1 = \left(\sum_{j \in \Lambda^+} \bar{x}_j \bar{x}_j' \right)^{-1} \sum_{j \in \Lambda^+} \bar{x}_j' \bar{y}_j .$$

ii) Adjusted Stage

If x_i is continuous, the effects of group sizes cannot be ignored, especially in high-dimensional spaces. Therefore, the adjustment stage is necessary. We define the adjusted dependent variable $y_{ij}(\hat{w}_1)$ as $y_{ij}(\hat{w}_1) = (x_{ij} - \bar{x}_j)' \hat{w}_1$ for $i \in I_j$.

We update the values of $\bar{y}_j$ to $\bar{y}_j^* =$ median of $\{y_{ij}(\hat{w}_1)\}$ for $j \in \Lambda^+$. We consider the regression model given by:

$$y_j(\hat{w}_1) = \bar{x}_j' w + e_j^*, \quad j \in \Lambda^+. \tag{4}$$

Let $\hat{w}_2$ be the OLS estimator of Equation (4). Using $\hat{w}_2$, adjust the data again and continue the procedure until the process converges. Since the process becomes a contraction mapping, it converges to the fixed point of the mapping, $\hat{w}^+$, if the sizes of $\{S_j\}$ are sufficiently small. If the process does not converge, reduce the sizes of $\{S_j\}$. For details, see Nawata [39].

iii) Final Stage

If necessary, redivide S into m^* nonoverlapping and fixed groups $\{S_j^*\} = \{S_1^*, S_2^*, \cdots, S_{m^*}^*\}$. Define I_j^*, I_j^{*+}, n_j^{*+}, and n_j^* as before, based on $\{S_j^*\}$. We update the adjusted dependent variable as

$$y_{ij}^* = y_{ij}(\hat{w}^+) = (x_{ij} - \bar{x}_j)' \hat{w}^+ \quad \text{for} \quad i \in I_j^*. \tag{5}$$

In this case, we adjust values in all groups. Consider the index set $I_j^{0*} = \{i : y_{ij}(\hat{w}^+) \geq M_j^*\}$, where $M_j^* = \text{median}\{y_{ij}^* : i \in I_j^*\}$. Define

$$\bar{x}_j^* = \begin{cases} \sum_i x_{ij} / n_j \text{ and } \bar{y}_j^* = M_j^*, \text{ if } M_j^* > 0 \text{ and } I_j^{0*} = \varnothing, \\ \text{the value minimizing } x_{ij}' \hat{w}^+ \text{ over } i \in I_j, \text{ otherwise.} \end{cases} \tag{6}$$

This means that we use the median of $\{y_{ij}^*\}$ if $\max_i \{y_{ij}^* : i \in I_j^0\} < M_j^*$ and 0 otherwise.

Since $\{\bar{y}_1^*, \bar{y}_2^*, \cdots, \bar{y}_{m^*}^*\}$ take 0 or positive values like the original data $\{y_1, y_2, \cdots, y_n\}$, and the asymptotic distribution of the median is normal, w is estimated by the GAME $\hat{w}$, which maximizes

$$L^*(w, \sigma^2) = \prod_{\bar{y}_j^* > 0} \frac{\sqrt{n_j}}{\sigma} \phi\left[\sqrt{n_j}(\bar{y}_j^* - \bar{x}_j^{*\prime} w)/\sigma\right] \prod_{\bar{y}_j^* = 0} \Phi\left[-(\bar{y}_j^* - \bar{x}_j^{*\prime} w)/\sigma\right]. \tag{7}$$

Unlike binary cases [40] [41], we can obtain the asymptotic distribution of the estimator, given by

$$\sqrt{n}(\hat{w} - w_0) \to N\left(0, \frac{1}{4 f(0)^2} A^{-1}\right), \tag{8}$$

$$A = \operatorname{plim}_{n\to\infty} \sum_{\bar{y}_j^* > 0} \left(\frac{n_j}{n} \bar{x}_j^* \bar{x}_j^{*\prime}\right) = \operatorname{plim}_{n\to\infty} \left(-\frac{1}{n} \frac{\partial^2 \log L^*}{\partial w \partial w'}\bigg|_{\theta_0}\right).$$

Here, $f(\cdot)$ is the density function of u_i and $\theta_0' = (w_0', \sigma_0^2)$ denotes the true parameter values. Variance estimation can be carried out using standard statistical software packages.

3. Monte Carlo Experiments

We consider 2D images where S is the rectangle defined by $0 < x_1 \leq 1$ and $0 < x_2 \leq 1$. x_1 and x_2 are divided into 300 equidistant grid lines, yielding n = 90,000 points per image. Let $x_{1\kappa}$ denote the κ-th grid line in x_1 and $x_{2\ell}$ de-

note the ℓ-th grid line in x_2. The intersection of $x_{1\kappa}$ and $x_{2\ell}$ is denoted as $x_{\kappa\ell}$. We consider the models

$$y^*_{\kappa\ell} = w_0 + w_1 x_{1\kappa} + w_2 x_{2\ell} + u_{\kappa\ell} \quad \text{and} \quad y_{\kappa\ell} = \max\left(0, y^*_{\kappa\ell}\right). \tag{9}$$

The boundary of the region is given by

$$x_2 = w^*_0 + w^*_1 x_1 \quad \text{where} \quad w^*_0 = w_0/w_2 \quad \text{and} \quad w^*_1 = -w_1/w_2 \,. \tag{10}$$

The true parameter values of w_0, w_1, and w_2 are 0.0, 1.0, and −1.0, respectively.

We compare three estimators:

1) CTMLE (assuming normal error terms),

2) GAME considered in this paper, and

3) MPE (max pooling estimator), which uses the maximum values within groups.

In contrast to MPE, GAME is based on the asymptotic normality of median.

For the i.i.d. case, we consider two distributions:

Case 1: Standard normal distribution, and

Case 2: Quasi-Cauchy distribution.

Since the estimation procedure failed to converge in many trials under the standard Cauchy distribution, we use the quasi-Cauchy distribution, where the error terms are drawn from U (0.002, 0.998) as a fat-tailed distribution. The variance of the distribution is 101.7.

For the heteroscedastic distributions, we consider the following cases:

Case 3: Heteroscedastic distribution I, $u_{\kappa\ell} = \left[1\left(x_{1\kappa} < 0.5\right) + 2 * 1\left(x_{1\kappa} \geq 0.5\right)\right]\varepsilon$, and

Case 4: Heteroscedastic distribution II, $u_{\kappa\ell} = \left[2 * 1\left(x_{1\kappa} < 0.5\right) + 1\left(x_{1\kappa} \geq 0.5\right)\right]\varepsilon$.

$1(\cdot)$ is an indicator function that takes the value 1 if the argument is true and 0 otherwise, and ε is a standard normal random variable. For GAME and MPE, the sample space of (x_1, x_2), denoted by S, is divided such that each group contains nine intersection points determined by three neighboring grid lines of x_1 and x_2. The number of groups is 100 × 100 = 10,000. We perform 1000 trials using EViews 13 for each case. CLMLE and GAME can be efficiently estimated using standard programs without incurring significant computational cost.

Tables 1-4 present the Monte Carlo results. When the error terms follow the i.i.d. normal distribution (**Table 1**, Case 1), the biases, standard deviations (SDs), and mean squared errors (MSEs) of CTMLE are small. This outcome is quite reasonable, since CTMLE is not only consistent but also efficient in this case. The biases of GAME are very small, but the SDs and MSEs are larger than those of CTMLE. For MPE, the biases, SDs, and MSE are small for w_1 and w_2, but that for w_0 is large (1.485). **Figure 2** illustrates the boundaries of the two regions calculated from the true parameter values and the averages of w^*_0 and w^*_1 values. The boundaries obtained from the true parameters, CTMLE, and GAME are quite similar and nearly coincide. In contrast, the boundary given by MPE is almost parallel to the other lines.

Table 1. Standard normal distribution: Case 1.

	w_0	w_1	w_2	w_0^*	w_1^*
True	0.0	1.0	−1.0	0.0	1.0
CTMLE					
mean	−0.0006	0.9998	−0.9987	0.0005	1.0012
SD	0.0154	0.0189	0.0188	0.0121	0.0193
MSE	0.0001	0.0002	0.0002	0.0001	0.0003
GAME					
mean	−0.0005	1.0013	−1.0002	0.0003	1.0013
SD	0.0180	0.0216	0.0235	0.0160	0.0275
MSE	0.0002	0.0003	0.0003	0.0002	0.0005
PME					
mean	1.4849	1.0028	−1.0038	1.4797	0.9995
SD	0.0161	0.0211	0.0209	0.0238	0.0214
MSE	2.2051	0.0005	0.0005	2.1903	3.9988

SD: Standard deviation, MSE: Mean squared error.

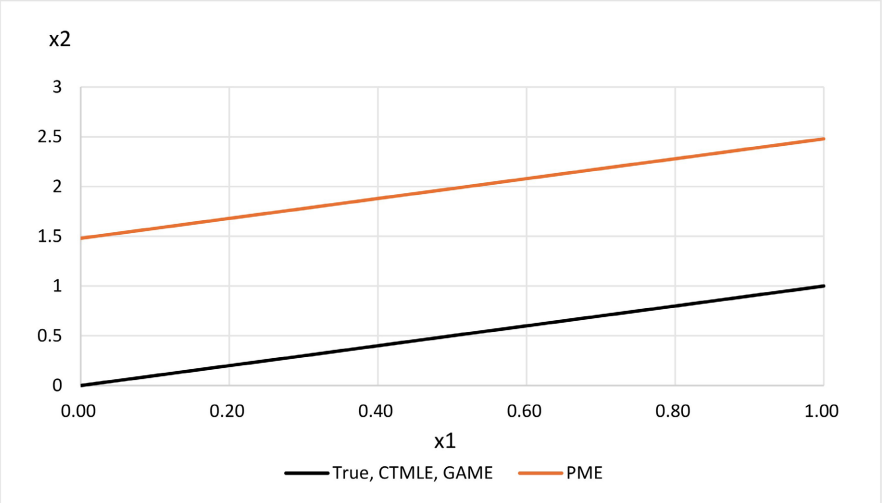

Figure 2. Boundaries obtained from the true parameter values (True), CTMLE, GAME, and PME for the standard normal distribution: Case 1 in **Table 1**.

When the error terms follow the quasi-Cauchy distribution (**Table 2**, Case 2), the biases of CTMLE are large, with values of 2.124, −0.818, and 0.096 for w_0, w_1, and w_2, respectively. In contrast, the biases, SDs, and MSEs of GAME are much smaller than those of CTMLE, with biases of −0.032, 0.073, and −0.075, respectively. MPE performs quite poorly, exhibiting not only very large biases but also very large SDs. **Figure 3** illustrates the boundaries obtained from the true parameter values and the averages of the estimated parameters. Both CTMLE and MPE produce incorrect boundaries; however, GAME yields an almost correct boundary much more accurate than that from CTMLE or MPE.

Table 2. Quasi-Cauchy distribution: Case 2.

	w_0	w_1	w_2	w_0^*	w_1^*
True	0.0	1.0	−1.0	0.0	1.0
CTMLE					
mean	2.1275	0.1816	−0.9039	2.5734	0.3329
SD	0.4941	0.6007	0.6575	3.4289	1.8383
MSE	4.7704	0.6699	0.4415	18.3794	3.8243
GAME					
mean	−0.0319	1.0731	−1.0745	−0.0299	0.9993
SD	0.1429	0.1707	0.1719	0.1390	0.1821
MSE	0.0214	0.0053	0.0351	0.0202	0.0332
PME					
mean	84.3342	−22.4730	−46.4981	−0.8000	−0.6911
SD	999.29	730.77	970.44	33.34	17.29
MSE	1005692.4	534582.4	943826.7	1112.2	301.8

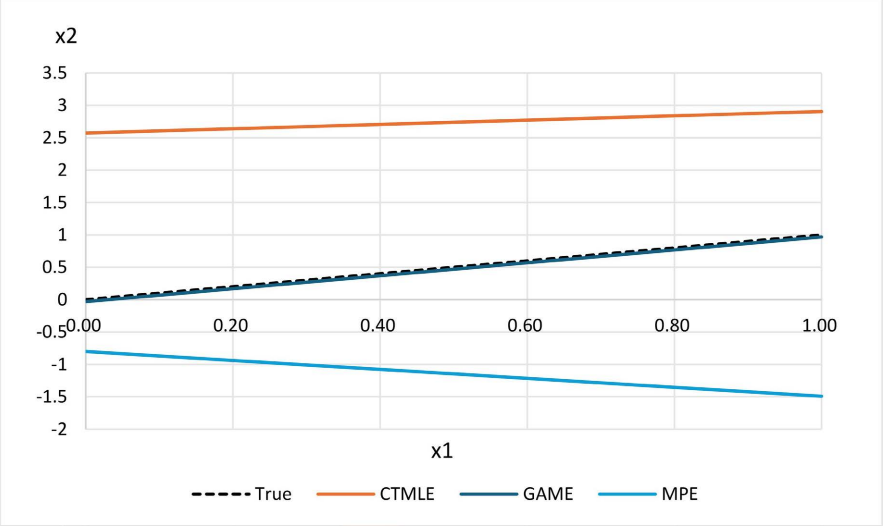

Figure 3. Boundaries obtained from the true parameter values (True), CTMLE, GAME, and PME for the quasi-Cauchy distribution: Case 2 in **Table 2**.

Table 3 and **Table 4** present the heteroscedastic distribution results. For Heteroscedastic distribution I (**Table 3**, Case 3), GAME clearly reduces biases for all parameters. The biases of MPE for w_0, w_1, and w_2 are 1.097, 2.248, and −0.019, respectively, which are large for w_0 and w_1 but relatively small for w_2. Since error term variances depend only on x_1, this may be reflected in the Monte Carlo results. **Figure 4** illustrates the boundaries calculated from the true parameter values and the estimates. The GAME boundary is closer to the true line than that from CTMLE, whereas MPE yields a poor result.

For Heteroscedastic distribution II (Table 4, Case 4), the biases and MSEs of GAME are smaller than those of CTMLE for all parameters. For MPE, although the bias for w_2 is small (−0.0017), the biases for w_0 and w_1 are large, at 3.339 and −2.299, respectively. Figure 5 shows the boundary lines of the two regions. GAME clearly improves upon CTMLE. As before, MPE performs poorly. In particular, $P[y_i > 0]$ decreases as x_1 increases, and the estimated target and background regions become opposite to the true ones in Case 4.

Table 3. Heteroscedastic distribution I: Case 3.

	w_0	w_1	w_2	w_0^*	w_1^*
True	0.0	1.0	−1.0	0.0	1.0
CTMLE					
mean	−0.5100	1.8759	−1.1269	−0.4528	1.6652
SD	0.0159	0.0228	0.0215	0.0202	0.0360
MSE	0.2603	0.7678	0.0166	0.2055	0.4438
GAME					
mean	−0.2574	1.4398	−1.1270	−0.2288	1.2783
SD	0.0208	0.0298	0.0306	0.0219	0.0389
MSE	0.0667	0.1943	0.0171	0.0528	0.0790
MPE					
mean	1.0967	3.2477	−1.0187	1.0772	3.1915
SD	0.0222	0.0338	0.0326	0.0234	0.1080
MSE	1.2033	5.0533	0.0014	1.1609	4.8144

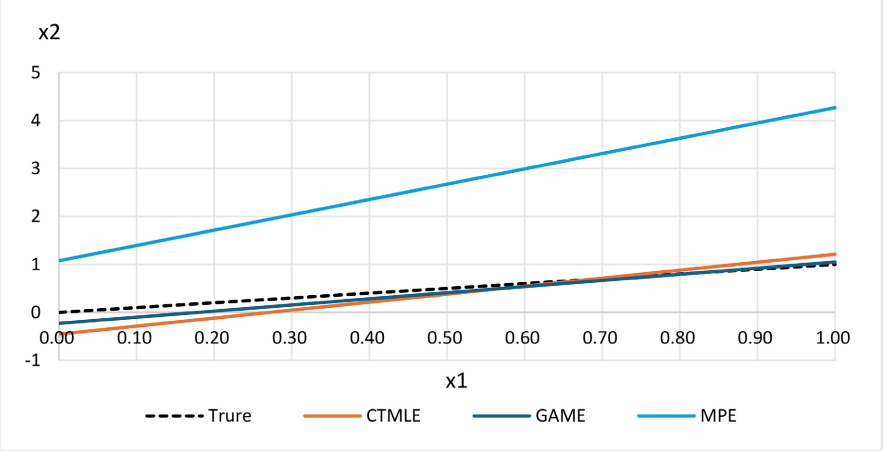

Figure 4. Boundaries obtained from the true parameter values (True), CTMLE, GAME, and PME for Heteroscedastic distribution I: Case 3 in Table 3.

Table 4. Heteroscedastic distribution II: Case 4.

	w_0	w_1	w_2	w_0^*	w_1^*
True	0.0	1.0	−1.0	0.0	1.0

Continued

CTMLE					
mean	0.4107	0.3129	−1.0375	0.3958	0.3017
SD	0.0154	0.0189	0.0188	0.0121	0.0193
MSE	0.1689	0.4724	0.0018	0.1568	0.4880
GAME					
mean	0.2010	0.6953	−0.9884	0.2032	0.7039
SD	0.0180	0.0216	0.0235	0.0160	0.0275
MSE	0.0407	0.0933	0.0007	0.0416	0.0884
PME					
mean	3.3389	−1.2292	−1.0017	3.3362	−1.2284
SD	0.0008	0.0011	0.0011	0.0094	0.0027
MSE	11.1490	4.9702	0.0011	11.1399	4.9684

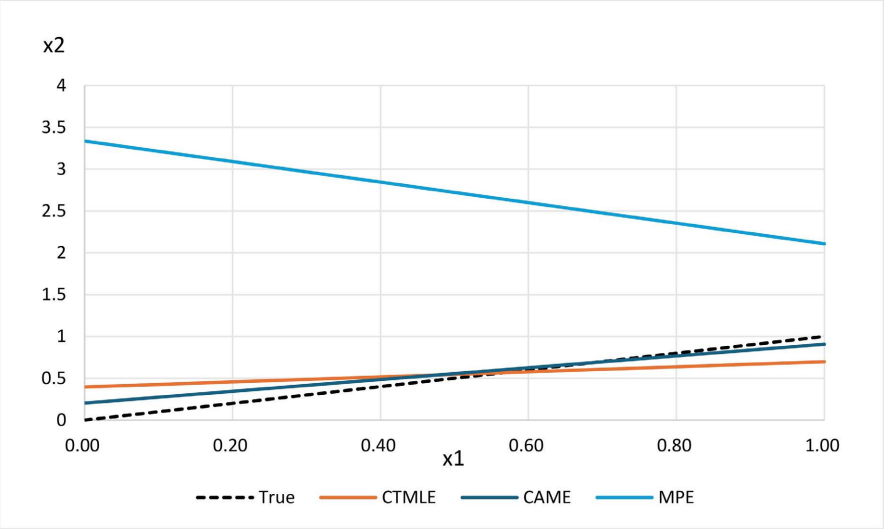

Figure 5. Boundaries obtained from the true parameter values (True), CTMLE, GAME, and PME for Heteroscedastic distribution II: Case 4 in **Table 4**.

4. Discussion

ReLU and max pooling are widely used in DL. However, reliable outcomes cannot be obtained if the distribution of error terms is misspecified or if inappropriate methods are applied. The Monte Carlo results support this claim. Both CTMLE and MPE exhibit substantial biases and large MSEs under non-normal (fat-tailed) or heteroscedastic distributions.

GAME is a semiparametric estimator and is consistent under very general assumptions. Although the SDs of GAME are slightly larger than those of CTMLE, GAME clearly outperforms CTMLE under non-normal (fat-tailed) and heteroscedastic distributions. When ε_i follows an i.i.d. standard normal distribution, the expected value of $\zeta = \max\{0, \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_9\}$ is 1.485, which coincides with the

bias of MPE for w_0. The biases for w_1 and w_2 are very small, at 0.003 and -0.004, respectively. The estimated boundary is nearly parallel to the true one. Many studies using ReLU exclude the constant term. This may be related to the fact that the max pooling method yields useful results in standard cases. In the other situations, however, MPE is a poor estimator not only for the constant but also for the slope coefficients. MPE produces highly inaccurate boundaries. In particular, when the variance is a decreasing function of x_1 (Case 4), the boundary slope becomes negative, and the estimated region where $P[y_i > 0]$ is opposite to the true one.

Figure 6 illustrates an example of max, average, and median pooling, where the maximum, average, and median are calculated from nine cells using $\mathrm{ReLU}(\eta) = \max\{0, \eta\}$, with η following the standard Cauchy distribution. In this figure, no signal (**S**) exists; all non-zero values represent random noise (**N**), and the signal-to-noise ratio (**S/N**) is zero. Max pooling yields a very high value of 79, as if important information were contained in those cells. Average pooling [42]-[45] also produces relatively large values (9.33). In contrast, median pooling results in zero. Max pooling emphasizes large values in the observations. While this may be useful for detecting very weak signals in the sample space, it can lead to incorrect results when noise is substantial and **S/N** is low. As high- or ultra-high-resolution 2D or 3D images with considerable noise (small **S/N**) are becoming increasingly common, careful treatment to eliminate or reduce noise is essential. In DL application areas, such as medical or satellite imaging where high noise is common and robustness is most critical, the approach considered in this paper may be quite useful.

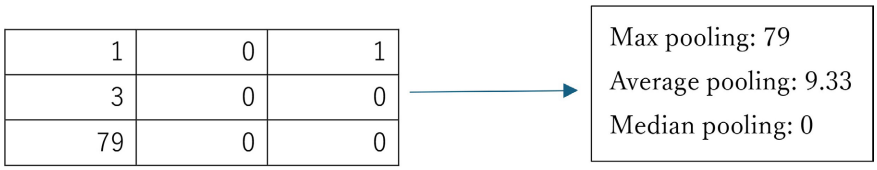

Figure 6. Example of max, average, and median pooling under the Cauchy distribution.

5. Conclusions

The ReLU and max pooling methods are widely used in DL. The Tobit model, commonly applied in econometrics and statistics, is closely related to ReLU. However, the knowledge obtained from the Tobit model has not been fully incorporated into DL. In this paper, we considered ReLU estimation using the Tobit framework combined with a grouping approach. The CTMLE, which assumes i.i.d. normal errors, exhibits large biases under fat-tailed (quasi-Cauchy) or heteroscedastic error distributions. We considered GAME, which combines grouping, adjustment, the median, and a weighted Tobit method. GAME is robust, remaining consistent under both non-normal and heteroscedastic error distributions. Moreover, unlike in binary cases, its asymptotic distribution is obtainable.

Monte Carlo results show that GAME outperforms CTMLE under non-normal

or heteroscedastic errors. In the i.i.d. normal case, where CTMLE is efficient, GAME's loss of efficiency is small; its biases and SDs remain close to those of CTMLE. In contrast, the MPE performs poorly, exhibiting large biases, SDs, and MSEs. In some cases, the region MPE suggests for $P[y_i > 0]$ is even the opposite of the true region. Max pooling selects the maximum value in each group, which can help detect weak signals but also amplifies noise when the signal-to-noise ratio is low. Thus, max pooling requires caution, and noise-reducing methods such as GAME are essential.

Acknowledgements

The author would also like to thank an anonymous reviewer for his/her helpful comments and suggestions.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Bai, Y. (2022) ReLU-function and Derived Function Review. *SHS Web of Conferences*, **144**, Article 02006. https://doi.org/10.1051/shsconf/202214402006

[2] Li, G.H.Y., Sekine, R., Nehra, R., Gray, R.M., Ledezma, L., Guo, Q., *et al.* (2022) All-optical Ultrafast ReLU Function for Energy-Efficient Nanophotonic Deep Learning. *Nanophotonics*, **12**, 847-855. https://doi.org/10.1515/nanoph-2022-0137

[3] Nworu, C.C., Ekpenyong, E.J., Chisimkwuo, J., Onyeukwu, C.N., Okwara, G. and Agwu, O.J. (2022) The Effects of Modified ReLU Activation Functions in Image Classification. *Biomedical Engineering and Medical Devices*, **7**, Article 1000237.

[4] Cabanilla, K.I.M., Mohammad, R.Z. and Lope, J.E.C. (2024) Neural Networks with ReLU Powers Need Less Depth. *Neural Networks*, **172**, Article 106073. https://doi.org/10.1016/j.neunet.2023.12.027

[5] Hu, P., Sun, L., Hu, C., Dai, L., Guo, S. and Yu, M. (2024) DReP: Deep ReLU Pruning for Fast Private Inference. *Journal of Systems Architecture*, **152**, Article 103156. https://doi.org/10.1016/j.sysarc.2024.103156

[6] Sooksatra, K., Hamerly, G. and Rivas, P. (2023) Is ReLU Adversarially Robust? arXiv:2405.03777.

[7] Ahmadnejad, A., Assadi, M.M. and Koohi, S. (2025) All-Optical Doubly Resonant Cavities for ReLU Function in Nanophotonic Deep Learning. arXiv:2504.19692v1.

[8] Horuz, C.C., Kasenbacher, G., Higuchi, S., Kairat, S., Stoltz, J., Pesl, M., *et al.* (2025) The Resurrection of the ReLU. arXiv:2505.22074v1.

[9] Mestre, J.I., Barrachina, S., Quezada, D. and Dolz, M.F. (2025) Deep Learning Inference Optimisation for IoT: Conv2D-ReLU-BN Layer Fusion and Quantisation. *The Journal of Supercomputing*, **81**, Article No. 621. https://doi.org/10.1007/s11227-025-07107-y

[10] Ullah, S., Javed, A., Aljasem, M. and Saudagar, A.K.J. (2025) Eff-ReLU-Net: A Deep Learning Framework for Multiclass Wound Classification. *BMC Medical Imaging*, **25**, Article No. 257. https://doi.org/10.1186/s12880-025-01785-z

[11] Ueno, Y., Hironaka, Y., Yoshikawa, N. and Yamanashi, Y. (2025) High-Speed, Ultra-Low-Power, and Robust Superconductive Neuron with ReLU Activation. *Neuromor-*

phic Computing and Engineering, **5**, Article 044003.
https://doi.org/10.1088/2634-4386/ae0aee

[12] Frei, S., Vardi, G., Bartlett, P.L., Srebro, N. and Hu, W. (2022) Implicit Bias in Leaky ReLU Networks Trained on High-Dimensional Data. arXiv:2210.07082.

[13] Mujhid, A., Surono, S., Irsalinda, N. and Thobirin, A. (2022) Comparison and Combination of Leaky ReLU and ReLU Activation Function and Three Optimizers on Deep CNN for COVID-19 Detection. In: Tallón-Ballesteros, A.J., Ed., *Frontiers in Artificial Intelligence and Applications*, IOS Press, 50-57.
https://doi.org/10.3233/faia220369

[14] Mary Shyni, H. and Chitra, E. (2024) PulmonNet V1: Leveraging the Benefit of Leaky ReLU Activation for the Local and Multi-Scale Global Feature Integration of Chest Radiographs to Classify Pulmonary Diseases. *Biomedical Signal Processing and Control*, **96**, Article 106600. https://doi.org/10.1016/j.bspc.2024.106600

[15] Singh, A.K. and Kumar, A. (2025) Multi-Objective: Hybrid Particle Swarm Optimization with Firefly Algorithm for Feature Selection with Leaky ReLU. *Discover Artificial Intelligence*, **5**, Article No. 192. https://doi.org/10.1007/s44163-025-00428-0

[16] Greene, W.H. (2020) Econometric Analysis. 8th and International Edition, Peason Education.

[17] Carson, R.T. and Sun, Y. (2007) The Tobit Model with a Non-Zero Threshold. *The Econometrics Journal*, **10**, 488-502. https://doi.org/10.1111/j.1368-423x.2007.00218.x

[18] Amore, M.D. and Murtinu, S. (2019) Tobit Models in Strategy Research: Critical Issues and Applications. *Global Strategy Journal*, **11**, 331-355.
https://doi.org/10.1002/gsj.1363

[19] Xue, L., Qu, A., Guo, X. and Hao, C. (2024) Research on Environmental Performance Measurement and Influencing Factors of Key Cities in China Based on Super-Efficiency SBM-Tobit Model. *Sustainability*, **16**, Article 4792.
https://doi.org/10.3390/su16114792

[20] Tobin, J. (1958) Estimation of Relationships for Limited Dependent Variables. *Econometrica*, **26**, 24-36. https://doi.org/10.2307/1907382

[21] Gholamalinezhad, H. and Khosravi, H. (2020) Pooling Methods in Deep Neural Networks, a Review. arXiv:2009.07485.

[22] Saeedan, F., Weber, N., Goesele, M. and Roth, S. (2018) Detail-Preserving Pooling in Deep Networks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 9108-9116.
https://doi.org/10.1109/cvpr.2018.00949

[23] Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., *et al.* (2022) A Comparison of Pooling Methods for Convolutional Neural Networks. *Applied Sciences*, **12**, Article 8643. https://doi.org/10.3390/app12178643

[24] Baniata, L.H., Kang, S., Alsharaiah, M.A. and Baniata, M.H. (2024) Advanced Deep Learning Model for Predicting the Academic Performances of Students in Educational Institutions. *Applied Sciences*, **14**, Article 1963.
https://doi.org/10.3390/app14051963

[25] Hosny, K.M., Magdi, A., ElKomy, O. and Hamza, H.M. (2024) Digital Image Watermarking Using Deep Learning: A Survey. *Computer Science Review*, **53**, Article 100662. https://doi.org/10.1016/j.cosrev.2024.100662

[26] Zhao, L. and Zhang, Z. (2024) A Improved Pooling Method for Convolutional Neural Networks. *Scientific Reports*, **14**, Article No. 1589.
https://doi.org/10.1038/s41598-024-51258-6

[27] Murray, N. and Perronnin, F. (2014) Generalized Max Pooling. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 2473-2480. https://doi.org/10.1109/cvpr.2014.317

[28] Brutzkus, A. and Globerson, A. (2021) An Optimization and Generalization Analysis for Max-Pooling Networks. *Proceedings of Machine Learning Research*, **161**, 1650-1660.

[29] Matoba, K., Dimitriadis, N. and Fleuret, F. (2023) Benefits of Max Pooling in Neural Networks: Theoretical and Experimental Evidence. *Transactions on Machine Learning Research*.

[30] Arulananth, T.S., Prakash, S.W., Ayyasamy, R.K., Kavitha, V.P., Kuppusamy, P.G. and Chinnasamy, P. (2024) Classification of Paediatric Pneumonia Using Modified Densenet-121 Deep-Learning Model. *IEEE Access*, **12**, 35716-35727. https://doi.org/10.1109/access.2024.3371151

[31] Mian, X., Bingtao, Z., Shiqiang, C. and Song, L. (2024) MCMP-Net: MLP Combining Max Pooling Network for sEMG Gesture Recognition. *Biomedical Signal Processing and Control*, **90**, Article 105846. https://doi.org/10.1016/j.bspc.2023.105846

[32] Kohler, M. and Langer, S. (2025) Statistical Theory for Image Classification Using Deep Convolutional Neural Network with Cross-Entropy Loss under the Hierarchical Max-Pooling Model. *Journal of Statistical Planning and Inference*, **234**, Article 106188. https://doi.org/10.1016/j.jspi.2024.106188

[33] Song, W., Dai, S., Huang, D., Song, J. and Antonio, L. (2021) Median-Pooling Grad-Cam: An Efficient Inference Level Visual Explanation for CNN Networks in Remote Sensing Image Classification. In: Lokoč, J., *et al.*, Eds., *Lecture Notes in Computer Science*, Springer International Publishing, 134-146. https://doi.org/10.1007/978-3-030-67835-7_12

[34] Nawata, K. (1990) Robust Estimation Based on Grouped-Adjusted Data in Censored Regression Models. *Journal of Econometrics*, **43**, 337-362. https://doi.org/10.1016/0304-4076(90)90124-c

[35] Bishop, C.M. and Bishop, H. (2024) Deep Learning: Foundation and Concepts. Springer.

[36] Segeth, K. (2021) Multivariate Data Fitting Using Polyharmonic Splines. *Journal of Computational and Applied Mathematics*, **397**, Article 113651. https://doi.org/10.1016/j.cam.2021.113651

[37] Dantony, E., Uhry, Z., Fauvernier, M., Coureau, G., Mounier, M., Trétarre, B., *et al.* (2024) Multidimensional Penalized Splines for Survival Models: Illustration for Net Survival Trend Analyses. *International Journal of Epidemiology*, **53**, dyae033. https://doi.org/10.1093/ije/dyae033

[38] Ameniya, T. (1985) Advanced Econometrics. Harvard University Press.

[39] Nawata, K. (1990) Robust Estimation Based on Grouped-Adjusted Data in Linear Regression Models. *Journal of Econometrics*, **43**, 317-336. https://doi.org/10.1016/0304-4076(90)90123-b

[40] Nawata, K. (1994) Estimation of the Boundary of the Two Regions by the Grouping Method. *Journal of the Japan Statistical Society*, **24**, 14-35.

[41] Nawata, K. (2022) An Analysis of Two-Dimensional Image Data Using a Grouping Estimator. *Open Journal of Statistics*, **12**, 33-48. https://doi.org/10.4236/ojs.2022.121003

[42] Bieder, F., Sandkühler, R. and Cattin, P.C. (2021) Comparison of Methods Generalizing Max- and Average-Pooling. arXiv:2103.01746.

[43] Yang, J., Xie, F., Fan, H., Jiang, Z. and Liu, J. (2018) Classification for Dermoscopy Images Using Convolutional Neural Networks Based on Region Average Pooling. *IEEE Access*, **6**, 65130-65138. https://doi.org/10.1109/access.2018.2877587

[44] Lee, C., Shih, J., Su, W., Lien, C. and Han, C. (2024) Combination of Global Maximum Pooling and Local Average Pooling for Unsupervised Fine-Grained Image Retrieval. *Proceedings of the* 2024 7*th Artificial Intelligence and Cloud Computing Conference*, Tokyo, 14-16 December 2024, 299-307. https://doi.org/10.1145/3719384.3719427

[45] Özdemir, C. (2023) Avg-Topk: A New Pooling Method for Convolutional Neural Networks. *Expert Systems with Applications*, **223**, Article 119892. https://doi.org/10.1016/j.eswa.2023.119892