An Alternative Probabilistic Interpretation of the Huber Loss

Gregory P. Meyer
Uber Advanced Technologies Group
gmeyer@uber.com

Abstract

The Huber loss is a robust loss function used for a wide range of regression tasks. To utilize the Huber loss, a parameter that controls the transitions from a quadratic function to an absolute value function needs to be selected. We believe the standard probabilistic interpretation that relates the Huber loss to the Huber density fails to provide adequate intuition for identifying the transition point. As a result, a hyper-parameter search is often necessary to determine an appropriate value. In this work, we propose an alternative probabilistic interpretation of the Huber loss, which relates minimizing the loss to minimizing an upper-bound on the Kullback-Leibler divergence between Laplace distributions, where one distribution represents the noise in the ground-truth and the other represents the noise in the prediction. In addition, we show that the parameters of the Laplace distributions are directly related to the transition point of the Huber loss. We demonstrate, through a toy problem, that the optimal transition point of the Huber loss is closely related to the distribution of the noise in the ground-truth data. As a result, our interpretation provides an intuitive way to identify well-suited hyper-parameters by approximating the amount of noise in the data, which we demonstrate through a case study and experimentation on the Faster R-CNN and RetinaNet object detectors.

1. Introduction

A typical problem in machine learning is estimating a function F_θ that maps from $x \in \mathbb{R}^n$ to $y \in \mathbb{R}$ given a set of training examples $\mathcal{D} = \{x_i, y_i\}_{i=0}^N$. The parameters of the function θ are often determined by minimizing a loss function $\mathcal{L}$,

$$\hat{\theta} = \arg\min_\theta \sum_{i=0}^N \mathcal{L}(y_i - F_\theta(x_i)) \tag{1}$$

and the choice of loss function can be crucial to the performance of the model. The Huber loss is a robust loss function that behaves quadratically for small residuals and linearly for large residuals [9]. The loss function was proposed over a half-century ago, and it is still widely used today for a variety of regression tasks, including 2D object detection [4, 14, 16, 18], 3D object detection [2, 3, 10, 22], shape and pose estimation [6, 11, 20], and stereo estimation [1].

A challenge with utilizing the Huber loss in practice is selecting an appropriate value to transition from a quadratic error to a linear error. Under certain assumptions, minimizing a loss function can be interpreted as maximizing the likelihood of y_i given x_i,

$$\hat{\theta} = \arg\max_\theta \prod_{i=0}^N p(y_i | x_i, \theta) \tag{2}$$

when $p(y_i|x_i,\theta) \propto \exp\left[-\mathcal{L}(y_i - F_\theta(x_i))\right]$. Therefore, the estimate $\hat{\theta}$ that minimizes the Huber loss can be interpreted as the maximum likelihood estimate of θ when $p(y_i|x_i,\theta)$ is the Huber density [9]. The Huber density can be difficult to interpret; as a result, hyper-parameter search is often employed to identify a satisfactory transition point for the Huber loss.

In this work, we propose an alternative probabilistic interpretation of the Huber loss. Our interpretation assumes y_i is a noisy estimate of the true value y_i^*, and we show that minimizing the Huber loss is equivalent to minimizing an upper-bound on the Kullback-Leibler (KL) divergence,

$$\sum_{i=0}^N D\left(p(y_i^*|y_i) \| q(y_i^*|x_i,\theta)\right) \tag{3}$$

when $p(y_i^*|y_i)$ and $q(y_i^*|x_i,\theta)$ are Laplace distributions and the scale of the distributions are directly related to the transition point of the Huber loss. For real-world problems, the value of y_i corresponding to x_i is often provided by a human annotator; therefore, it is likely to contain some amount of noise. We believe that approximating the amount of noise in the ground-truth is a more intuitive way to determine the transition point for the Huber loss than reasoning about the Huber density.

In the following sections, we survey the related work (Section 2), review the Huber loss and maximum likelihood estimation in detail (Section 3), propose our alternative probabilistic interpretation of the Huber loss (Section 4), utilize a toy problem to illustrate the relationship between the optimal transition point of the Huber loss and the noise distribution of the ground-truth (Section 5), leverage our interpretation to analyze the loss functions utilized by modern object detectors (Section 6), and show that our proposed interpretation can lead to better hyper-parameters (Section 7).

2. Related Work

Noy and Crammer [17], remarked on the similarity between the Huber loss and the KL divergence of Laplace distributions, which motivates their use of a Laplace-like family of distributions in the PAC-Bayes framework. However, they did not explore the relationship beyond this observation. In this work, we further pursue the connection between the Huber loss and the KL divergence of Laplace distributions, and we identify the links between the parameters of the Huber loss and the parameters of the Laplace distributions.

Lange [12], proposed a set of potential functions for image reconstruction that behave like the Huber loss, but unlike the Huber loss, these functions are more than once differentiable. In this work, we propose a loss function which is similar to a potential function in [12]. However, our proposed loss is derived directly from the KL divergence of Laplace distributions; whereas, the potential functions in [12] are derived through double integration of symmetric and positive functions.

3. Background

3.1. Huber Loss

Loss functions commonly used for regression are $L_1(x) = |x|$ and $L_2(x) = \frac{1}{2}x^2$. Both of these functions have advantages and disadvantages; L_1 is less sensitive to outliers in the data, but it is not differentiable at zero. Whereas, the L_2 is differentiable everywhere, but it is highly sensitive to outliers. Huber proposed the following loss as a compromise between the L_1 and L_2 losses [9]:

$$H_\alpha(x) = \begin{cases} \frac{1}{2}x^2, & |x| \leq \alpha \\ \alpha\left(|x| - \frac{1}{2}\alpha\right), & |x| > \alpha \end{cases} \tag{4}$$

where $\alpha \in \mathbb{R}^+$ is a positive real number that controls the transition from L_1 to L_2. The Huber loss is both differentiable everywhere and robust to outliers.

A disadvantage of the Huber loss is that the parameter α needs to be selected. In this work, we propose an intuitive and probabilistic interpretation of the Huber loss and its parameter α, which we believe can ease the process of hyper-parameter selection. Next, we review how minimizing the loss functions are related to maximum likelihood estimation.

3.2. Maximum Likelihood Estimation

Assume we have some data $\mathcal{D} = \{x_i, y_i\}_{i=0}^N$ independently drawn from some unknown distribution. Let us model the relationship between x_i and y_i as

$$y_i = F_\theta(x_i) + \epsilon \tag{5}$$

where F_θ is a deterministic function parameterized by θ, and ϵ is random noise drawn from some known distribution. The goal of maximum likelihood estimation is to identify the parameter $\hat{\theta}$ that maximizes the likelihood of y_i given x_i across the dataset $\mathcal{D}$. Note that maximizing the likelihood of y_i given x_i is equivalent to minimizing the negative log likelihood,

$$\begin{aligned} \hat{\theta} &= \arg\max_\theta \prod_{i=0}^N p(y_i|x_i, \theta) \\ &= \arg\min_\theta -\sum_{i=0}^N \log p(y_i|x_i, \theta). \end{aligned} \tag{6}$$

Consider the case when the noise ϵ is drawn independently from a zero-mean Gaussian distribution. The probability density for y_i given x_i becomes

$$p(y_i|x_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - F_\theta(x_i))^2}{2\sigma^2}\right) \tag{7}$$

where $\sigma \in \mathbb{R}^+$ is the standard deviation of the noise, and the negative log likelihood becomes

$$-\log p(y_i|x_i, \theta) = \log\sqrt{2\pi\sigma^2} + \frac{(y_i - F_\theta(x_i))^2}{2\sigma^2}. \tag{8}$$

Notice that

$$\begin{aligned} \hat{\theta} &= \arg\min_\theta -\sum_{i=0}^N \log p(y_i|x_i, \theta) \\ &= \arg\min_\theta \sum_{i=0}^N (y_i - F_\theta(x_i))^2 \end{aligned} \tag{9}$$

by assuming a constant σ and dropping the constant term. Therefore, identifying $\hat{\theta}$ that minimizes the L_2 loss over the dataset is equivalent to the maximum likelihood estimate of θ when $p(y_i|x_i, \theta)$ follows a Gaussian distribution. In addition, minimizing the L_1 loss can be shown to be the same as the maximum likelihood estimation when the noise is drawn from a Laplace distribution. In [9], it is demonstrated that minimizing the Huber loss provides the maximum likelihood estimate when the probability density takes the form $p(y_i|x_i, \theta) \propto \exp\left[-H_\alpha(y_i - F_\theta(x_i))\right]$, which is sometimes referred to as the Huber density. The Huber loss is a combination of the L_1 and L_2 losses; therefore, the Huber density is a hybrid of the Gaussian and Laplace distributions.

The Huber density is more complicated than either the Gaussian or Laplace distribution individually, and we believe this complexity makes it challenging to use this interpretation of the Huber loss for selecting the parameter α. For this reason, we propose an alternative probabilistic interpretation.

4. Proposed Method

Like above, assume we have a dataset $\mathcal{D} = \{x_i, y_i\}_{i=0}^{N}$, but let us consider the following relationships:

$$y_i^* = y_i + \epsilon_1 \tag{10}$$

$$y_i^* = F_\theta(x_i) + \epsilon_2 \tag{11}$$

where y_i^* is an unknown value we would like to estimate with $F_\theta(x_i)$, y_i is a known estimate of y_i^*, and ϵ_1 and ϵ_2 are random noise variables drawn independently from separate but known distributions. Since y_i^* is hidden, we are unable to estimate $\hat{\theta}$ by directly maximizing the likelihood of y_i^* given x_i. Alternatively, we can estimate $\hat{\theta}$ by minimizing the Kullback-Leibler (KL) divergence between the distributions $p(y_i^*|y_i)$ and $q(y_i^*|x_i, \theta)$. Intuitively, $p(y_i^*|y_i)$ represents our uncertainty in the label y_i, and $q(y_i^*|x_i, \theta)$ represents our uncertainty in the model's prediction $F_\theta(x_i)$. Also, note that minimizing the KL divergence is equivalent to minimizing the cross entropy,

$$\begin{aligned}\hat{\theta} &= \arg\min_\theta \sum_{i=0}^{N} D\left(p(y_i^*|y_i) \| q(y_i^*|x_i, \theta)\right) \\ &= \arg\min_\theta -\sum_{i=0}^{N} \left(\int_{-\infty}^{\infty} p(y_i^*|y_i) \log q(y_i^*|x_i, \theta) dy_i^* \right)\end{aligned} \tag{12}$$

since the entropy of $p(y_i^*|y_i)$ is constant. If $p(y_i^*|y_i)$ is a Dirac delta function centered on y_i, i.e. the label contains zero noise, minimizing the cross entropy is equivalent to minimizing the negative log likelihood of $q(y_i|x_i, \theta)$. Therefore, finding $\hat{\theta}$ by minimizing the KL divergence is exactly the maximum likelihood estimate of θ when $y_i^* = y_i$.

Let us assume both the labels and the predictions are contaminated with outliers, i.e. both ϵ_1 and ϵ_2 are drawn from Laplace distributions. The corresponding probability densities are

$$p(y_i^*|y_i) = \frac{1}{2b_1} \exp\left(-\frac{|y_i^* - y_i|}{b_1}\right) \tag{13}$$

and

$$q(y_i^*|x_i, \theta) = \frac{1}{2b_2} \exp\left(-\frac{|y_i^* - F_\theta(x_i)|}{b_2}\right) \tag{14}$$

where $b_1 \in \mathbb{R}^+$ and $b_2 \in \mathbb{R}^+$ define the scale of the label uncertainty and prediction uncertainty, respectively. Furthermore, the KL divergence becomes

$$\begin{aligned}&D\left(p(y_i^*|y_i) \| q(y_i^*|x_i, \theta)\right) \\ &\quad = \frac{b_1 \exp\left(-\frac{|y_i - F_\theta(x_i)|}{b_1}\right) + |y_i - F_\theta(x_i)|}{b_2} + \log\frac{b_2}{b_1} - 1\end{aligned} \tag{15}$$

by integrating over all values of y_i^*. For a derivation, please refer to Appendix A. In the following sections, we propose a loss derived from the KL divergence of Laplace distributions, show that it is related to the Huber loss, and use the relationship to gain further insight into the Huber loss.

4.1. Proposed Loss Function

We propose the following loss function:

$$D_{\alpha,\beta}(x) = \frac{\alpha \exp\left(-\frac{|x|}{\alpha}\right) + |x| - \alpha}{\beta} \tag{16}$$

which is derived from the KL divergence of Laplace distributions (Equation (15)) by removing the existing constant terms and by adding a new constant term to ensure the minimum value is always zero. The variable x is equal to the difference in the means of the Laplace distributions. The parameter $\alpha \in \mathbb{R}^+$ directly corresponds to the scale of the noise in the label (b_1), and $\beta \in \mathbb{R}^+$ corresponds to the scale of the noise in the prediction (b_2). As a result, the parameters, α and β, have an intuitive and probabilistic interpretation related to the variance of the Laplace distributions. Since our modification to Equation (15) simply removes the constant penalty for the mismatch in the standard deviation of the distributions, our loss function is identical to the KL divergence when $b_1 = b_2 = \alpha = \beta$.

4.2. Relationship to the Huber Loss

To demonstrate the relationship between our proposed loss and the Huber loss, let us start by considering the behavior of our loss function when $|x|$ is small with respect to α. The second-order approximation of Equation (16) about zero is

$$D_{\alpha,\beta}(x) \approx D_{\alpha,\beta}(0) + D'_{\alpha,\beta}(0)x + \frac{D''_{\alpha,\beta}(0)}{2}x^2 = \frac{1}{2\alpha\beta}x^2. \tag{17}$$

Refer to Appendix B, for a derivation of the derivatives and a proof of their existence at $x = 0$. Furthermore, when $|x|$ is large with respect to α the exponential term in Equation (16) goes to zero,

$$D_{\alpha,\beta}(x) \approx \frac{|x| - \alpha}{\beta}. \tag{18}$$

As a result, Equation (16) can be approximated using the following piecewise function:

$$D_{\alpha,\beta}(x) \approx \begin{cases} \frac{1}{2\alpha\beta}x^2, & |x| \leq \alpha \\ \frac{|x|-\alpha}{\beta}, & |x| > \alpha. \end{cases} \tag{19}$$

Like the Huber loss, our proposed loss behaves quadratically when the residual is small and linearly when the residual is large. In addition, the following configurations tightly bound the Huber loss:

$$D_{\alpha,{}^1\!/_\alpha}(x) \leq H_\alpha(x) \leq D_{{}^\alpha\!/_2,{}^1\!/_\alpha}(x). \tag{20}$$

The relationship between the loss functions is illustrated in Figure 1, and a formal proof of the bounds is provided in Appendix C.

Minimizing the Huber loss with parameter α is equivalent to minimizing an upper-bound on the KL divergence of two Laplace distributions when the scale of the label distribution $b_1 = \alpha$, and the scale of the prediction distribution $b_2 = {}^1\!/\alpha$. Conversely, minimizing the KL divergence of two Laplace distributions with $b_1 = {}^\alpha\!/_2$ and $b_2 = {}^1\!/_\alpha$ is equivalent to minimizing an upper-bound on the Huber loss with parameter α. We believe this alternative probabilistic interpretation of the Huber loss provides significant insight into the parameter α, which we demonstrate in the remaining sections.

4.3. Properties of the Loss Function

Notice that scaling x by a positive real number, $\gamma \in \mathbb{R}^+$, is equivalent to $D_{\alpha,\beta}(\gamma x) = D_{{}^\alpha\!/_\gamma,{}^\beta\!/_\gamma}(x)$ whereas scaling the loss by $\lambda \in \mathbb{R}^+$ is equivalent to $\lambda D_{\alpha,\beta}(x) = D_{\alpha,{}^\beta\!/_\lambda}(x)$. Both of these properties are trivial to show through algebraic manipulation.

In the following sections, we will analyze the Huber loss with the approximation $H_\alpha(x) \approx D_{\alpha,{}^1\!/_\alpha}(x)$. Combining the above properties with the approximation, we observe that

$$\lambda H_\alpha(\gamma x) \approx D_{{}^\alpha\!/_\gamma,{}^1\!/_{\alpha\gamma\lambda}}(x). \tag{21}$$

Therefore, scaling the input to the Huber loss is equivalent to inversely scaling the label and prediction distributions, and scaling the output is equivalent to inversely scaling the prediction distribution.

5. Toy Problem: Polynomial Fitting

Our proposed interpretation of the Huber loss suggests there is a relationship between the parameter of the loss and the scale of the label noise. We would like to determine whether or not knowing this relationship and having a good estimate for the noise can help us select a good parameter for the loss. We employ a toy problem, where we control the amount of label noise, to show that the optimal α parameter is closely related to the scale of the label noise.

For our toy problem, we fit a one-dimensional polynomial to a set of sample points. To create our training examples, $\mathcal{D} = \{x_i, y_i\}_{i=0}^N$, we randomly sample $x_i \in [-\delta, \delta]$ uniformly and generate its corresponding label as $y_i = F_{\theta^*}(x_i) + \epsilon$ where $\theta^* \in \mathbb{R}^D$ are the parameters of the ground-truth polynomial and ϵ is noise randomly drawn from a distribution we specify. To create our test set, we use the same process but with $\epsilon = 0$.

We estimate the parameters of the predicted polynomial, $\hat{\theta} \in \mathbb{R}^K$, by minimizing the sum of the Huber loss over the training examples. For many tasks, the exact model for the data is unknown, and the mismatch between the true and estimated model could be a source of noise in the predictions. For this reason, we set $K > D$.

To evaluate the estimated parameters, we compute the root mean square error (RMSE) over the test set. We use gradient descent to identity $\hat{\theta}$, and we use grid search to identify the optimal learning rate and α parameter.

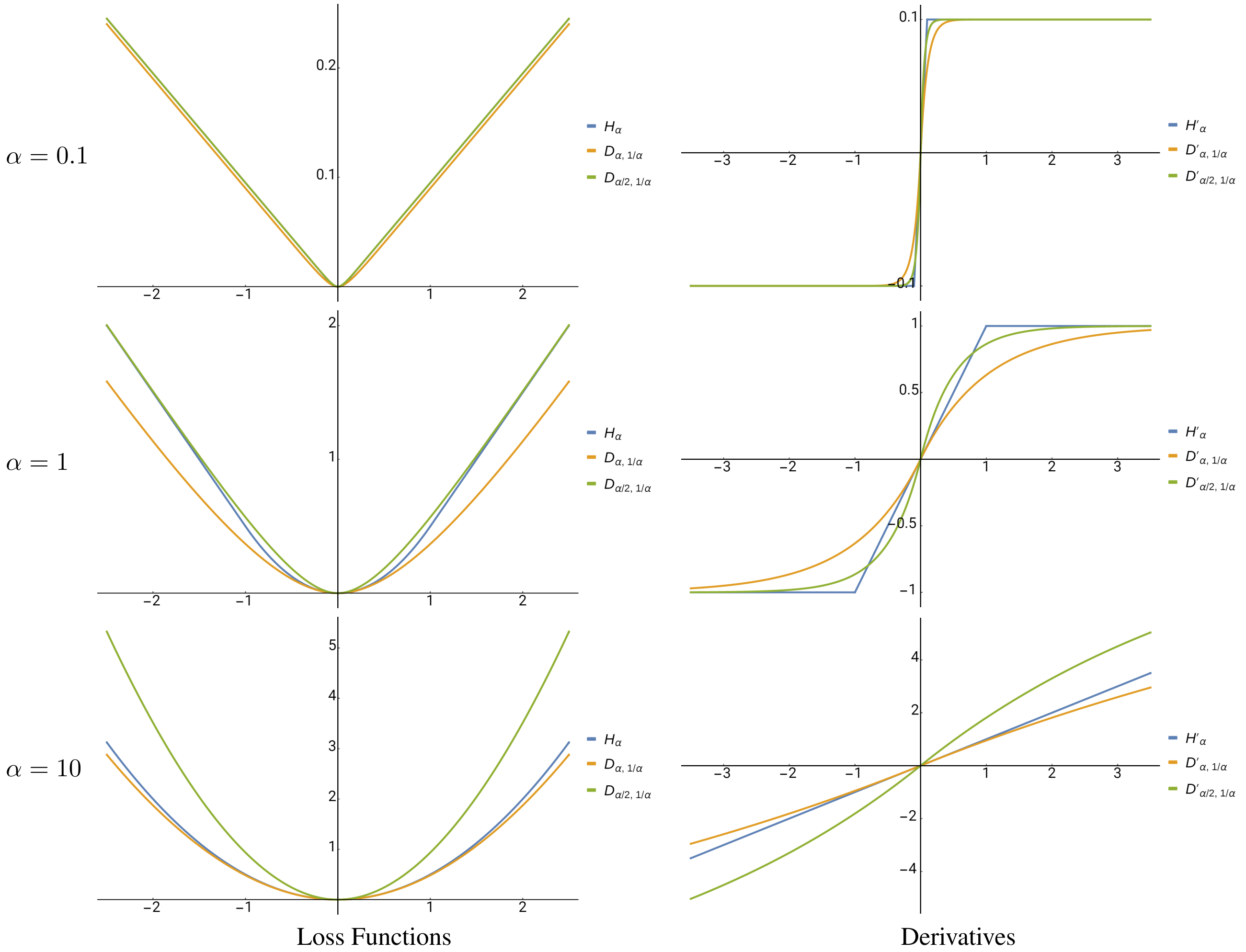
For each experiment, we sample $N = 10000$ points with $\delta = 2$. We arbitrarily select the parameters of our ground-truth polynomial as $\theta^* = [6, -3, -25, 15, 20, -10]$; therefore, $D = 6$ and $F_{\theta^*}(x) = 6x^5 - 3x^4 - 25x^3 + 15x^2 + 20x - 10$. Furthermore, we set $K = D + 2$. Since the Huber loss is designed to be robust to outliers, we sample ϵ from three different heavy-tailed distributions: the Laplace, Logistic, and Cauchy distributions. The results of our experiments are shown in Figure 2, which illustrates there is a near linear relationship between the scale of the label noise and the optimal α parameter for each of the distributions. This suggests that knowing or estimating the label noise can enable us to identify a suitable α parameter.

Next, we will demonstrate with a real-world problem that approximating the label noise when it is unknown is an intuitive and effective method for selecting well-suited hyper-parameters.

6. Case Study: Faster R-CNN

With our proposed interpretation, we analyze the loss functions used in a modern object detector, Faster R-CNN [18], which is arguably one of the most important advancements in object detection in recent history. Their work has inspired the development of several other object detectors including SSD [16], FPN [13], RetinaNet [14], and Mask R-CNN [7], all of which leverage the same loss functions for bounding box regression.

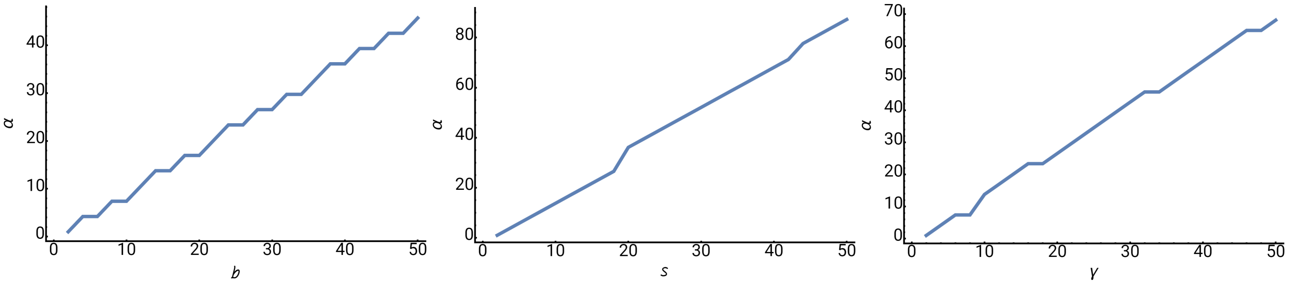
The Faster R-CNN network architecture consists of two primary parts, a region proposal network and an object detection network. The proposal network identifies regions that may contain objects, and the detection network refines and classifies the proposed regions. To regress a bounding box, both the proposal network and the detection network utilize the Huber loss. In their work, a bounding box is parameterized by its center and dimensions. Let us start by an-



Loss Functions

Derivatives

Figure 1: A comparison between the Huber loss (H_α) and our proposed loss ($D_{\alpha,\beta}$) derived from the KL divergence of Laplace distributions. The loss function H_α is lower-bounded by $D_{\alpha,{}^1\!/_\alpha}$ and upper-bounded by $D_{\alpha/_2,{}^1\!/_\alpha}$. The left column depicts the loss functions and the right column visualizes their derivatives. In addition, each row of the figure depicts a different set of hyper-parameters.



(a) Laplace Distribution

(b) Logistic Distribution

(c) Cauchy Distribution

Figure 2: The results of our toy problem. The x-axis is the parameter that controls the scale of the respective noise distributions and the y-axis is the optimal α parameter for the Huber loss. For each distribution, there is an approximate linear relationship between the noise and the optimal parameter.

alyzing the center prediction; the target for the x-coordinate of the center is

$$t_x^* = \frac{x^* - x_a}{w_a} \tag{22}$$

where x^* is the x-coordinate of the ground-truth center, x_a is the x-coordinate of the corresponding anchor, and w_a is the width of the anchor. A similar target is used for the center's y-coordinate except the height of the anchor is used instead of the width. For the proposal network, the anchors are predefined, whereas the detection network uses the proposals as its anchors.

In the paper, the authors state that they use $\lambda H_1(t_x - t_x^*)$ to penalize the model's prediction, t_x, during training where $\lambda = 10$ is a weighting parameter [18]. To interpret this loss, let us first re-write the residual in terms of the center displacement,

$$\begin{aligned} t_x - t_x^* &= t_x - \frac{x^* - x_a}{w_a} \\ &= \frac{(t_x w_a + x_a) - x^*}{w_a} = \frac{x - x^*}{w_a} \end{aligned} \tag{23}$$

where $x = t_x w_a + x_a$ is the predicted x-coordinate of the center. Utilizing Equation (21), we see that $\lambda H_1(t_x - t_x^*) \approx D_{w_a, w_a/\lambda}(x - x^*)$. Based on this interpretation, the scale of noise in the prediction is one-tenth the width of the anchor, but the scale of the label uncertainty is the full width of the anchor. Obviously, assuming the labels contain this amount of uncertainty is inappropriate. As it happens, the loss function and targets used in the current implementation of Faster R-CNN differ significantly from the paper [5]. Interpreting the implementation is important because it is the foundation for several other object detectors [7, 13, 14, 16].

In the implementation of Faster R-CNN [5], the authors scale the Huber loss by $1/\alpha$. Furthermore, the ground-truth targets have been shifted and scaled,

$$\tilde{t}_x^* = \frac{t_x^* - \mu_x}{\sigma_x} \tag{24}$$

by constant values $\mu_x \in \mathbb{R}$ and $\sigma_x \in \mathbb{R}^+$. Let us repeat our analysis with these modifications. Like before, we begin with re-writing the residual,

$$\begin{aligned} t_x - \tilde{t}_x^* &= t_x + \frac{\mu_x}{\sigma_x} - \frac{x^* - x_a}{\sigma_x w_a} \\ &= \frac{[(t_x \sigma_x + \mu_x)\, w_a + x_a] - x^*}{\sigma_x w_a} = \frac{\tilde{x} - x^*}{\sigma_x w_a} \end{aligned} \tag{25}$$

where $\tilde{x} = (t_x \sigma_x + \mu_x)\, w_a + x_a$. Next, let us consider the relationship between their loss function and our proposed loss function:

$$\frac{\lambda}{\alpha} H_\alpha(t_x - \tilde{t}_x^*) \approx D_{\alpha\sigma_x w_a, \sigma_x w_a/\lambda}(\tilde{x} - x^*). \tag{26}$$

With these additional complexities, the authors were unknowingly able to independently manipulate the scale of the label and prediction noise. To train the proposal network, $\lambda = 1$, $\alpha = 1/9$, and $\sigma_x = 1$, and to train the detection network $\lambda = 1$, $\alpha = 1$, and $\sigma_x = 1/10$. For both networks, the scale of the label noise is similar, a ninth and tenth of the anchor width, which is a much more reasonable assumption. With this interpretation, the scale of prediction uncertainty is significantly larger for the proposal network compared to the detection network, the full width of the anchor versus a tenth of the width. Intuitively, it makes sense to have a smaller prediction uncertainty for the detection network because it is designed to refine the output of the proposal network; however, a proposal uncertainty of this magnitude may be too extreme.

Likewise, we can perform the same analysis for the dimensions of the bounding box. The target for the width of the bounding box is

$$t_w^* = \log \frac{w^*}{w_a} \tag{27}$$

and there is a similar target for the height of the bounding box. As before, the target is shifted and scaled by $\mu_w \in \mathbb{R}$ and $\sigma_w \in \mathbb{R}^+$,

$$\tilde{t}_w^* = \frac{t_w^* - \mu_w}{\sigma_w}. \tag{28}$$

By re-writing the difference, we obtain the following:

$$\begin{aligned} t_w - \tilde{t}_w^* &= t_w - \frac{\log w^* - \log w_a - \mu_w}{\sigma_w} \\ &= \frac{(t_w \sigma_w + \mu_w + \log w_a) - \log w^*}{\sigma_w} \\ &= \frac{\log \tilde{w} - \log w^*}{\sigma_w} \end{aligned} \tag{29}$$

where $\tilde{w} = \exp(t_w \sigma_w + \mu_w)\, w_a$ is the predicted width of the bounding box. Since the log of the width can be difficult to interpret, let us consider the following approximation:

$$\log \frac{w^*}{w_a} \approx \frac{w^*}{w_a} - 1 \tag{30}$$

which is the first-order approximation of the logarithm when $w^*/w_a \approx 1$. This is not an outlandish assumption because the intersection-over-union (IoU) between the anchor and the ground-truth bounding box needs to be significant for the ground-truth to be matched with the anchor.[1] Now, the difference can be approximated as

$$\begin{aligned} t_w - \tilde{t}_w^* &\approx t_w + \frac{\mu_w + 1}{\sigma_w} - \frac{w^*}{\sigma_w w_a} \\ &\approx \frac{(t_w \sigma_w + \mu_w + 1)\, w_a - w^*}{\sigma_w w_a} \approx \frac{\tilde{w} - w^*}{\sigma_w w_a} \end{aligned} \tag{31}$$

[1] Refer to Appendix D for experimental validation of the target approximation.

where $\tilde{w} \approx (t_w \sigma_w + \mu_w + 1)\, w_a$, which conforms with the first-order approximation of the exponential function when $t_w \sigma_w + \mu_w \approx 0$. Leveraging our interpretation of the Huber loss, we observe

$$\begin{aligned} \frac{\lambda}{\alpha} H_\alpha(t_w - \tilde{t}^*_w) &\approx D_{\alpha\sigma_w, \sigma_w/\lambda}(\log \tilde{w} - \log w^*) \\ &\approx D_{\alpha\sigma_w w_a, \sigma_w w_a/\lambda}(\tilde{w} - w^*). \end{aligned} \tag{32}$$

In this case, $\lambda = 1$, $\alpha = {}^1\!/_9$, and $\sigma_w = 1$ for the proposal network, and $\lambda = 1$, $\alpha = 1$, and $\sigma_w = {}^1\!/_5$ for the detection network. Interestingly, the label noise is assumed to be higher for the detection network compared to the proposal network, which could be less than optimal.

It is unclear how the authors arrived at these peculiar hyper-parameters, undoubtedly through some form of parameter sweep. Based on our interpretation, we believe the hyper-parameters could be improved upon, which we demonstrate in the following section. In general, we believe that our interpretation can aid in hyper-parameter selection by eliminating inappropriate values.

7. Experiments

In this section, we perform experiments on Faster R-CNN as well as another modern object detector, RetinaNet. Our goal is not to obtain state-of-the-art object detection performance, there is a wealth of literature that improves upon these methods; instead, our goal is to demonstrate that our proposed interpretation of the Huber loss can lead to hyper-parameters better suited to the task of bounding box regression. Furthermore, our aim is not to replace the Huber loss with our proposed loss; rather, we want to leverage the relationship between the losses to gain insight into the Huber loss.[2] For these reasons, we limit our modifications to the following hyper-parameters: α, λ, σ_x, σ_y, σ_w, and σ_h (refer to Section 6 for more details).[3]

7.1. Faster R-CNN

To conduct our experiments, we utilize the implementation and framework provided by the authors of Faster R-CNN [5]. The deepest neural network supported by their framework is VGG-16 [19], and the largest dataset is MS-COCO 2014 [15]. For all of our experiments, we train the Faster R-CNN model with a VGG-16 backbone on the MS-COCO 2014 training set and measure the object detection performance on the validation set. The MS-COCO 2014 dataset [15] contains objects from 80 different classes, and it includes over 80k images for training and 40k images for validation. The metric used to measure object detection performance is the mean average precision (mAP) at various intersection-over-union (IoU) thresholds. To evaluate our experiments, we consider the mAP at 0.5 IoU and 0.75 IoU thresholds, as well as, the mAP averaged over 0.5-0.95 IoU thresholds. Unless otherwise stated, we use the default configurations set by the authors to train and test the models.

For our initial experiment, we train a model using the hyper-parameters as they are described in the publication [18]. Afterwards, we evaluate the parameters as they are specified in the current implementation of Faster R-CNN [5]. Lastly, we leverage our interpretation to propose three new sets of hyper-parameters. A full list of the parameters used as well as their corresponding interpretation is provided in Table 1 and Table 2, respectively. Notice for our proposed hyper-parameters, the label noise does not vary between the proposal and detection network, only the prediction noise varies. Based on our interpretation, we believe the label noise should not change between the two networks while the prediction noise should.

The results of the experiments are presented in Table 3. We were unable to exactly reproduce the results as they are listed in [18], likely due to changes made to the implementation by the authors that are unrelated to the hyper-parameters of the Huber loss. Regardless, in our experiments, the published hyper-parameters perform the worst by a significant margin, which should not be a surprise given our interpretation. The authors of Faster R-CNN were able to improve performance of the detector by tuning the hyper-parameters in the implementation [5]. We were able to further improve performance by reducing the estimated amount of noise in the labels and predictions. Specifically, we were able to raise performance at larger IoU thresholds. Achieving an improvement in mAP at higher thresholds requires more accurate bounding boxes; therefore, it makes sense that reducing the estimated uncertainty increases performance at those thresholds. Experiment A and B trade-off performance at 0.5 and 0.75 IoU, and Experiment C identifies a good balance between both. These results are significant because they were obtained by leveraging the intuition provided by our proposed interpretation of the Huber loss without the need for an exhaustive hyper-parameter search.

7.2. RetinaNet

As previously discussed, Faster R-CNN uses two networks or stages to perform object detection. Whereas, RetinaNet [14] uses only a single stage; therefore, it uses one network to regress the bounding box and classify the objects. For our experiments, we utilize the official implementation of RetinaNet [21]. In the RetinaNet implementation, the loss function utilized to regress bounding boxes is identical to the loss function used for the proposal network in the Faster R-CNN implementation [5]. For this reason, we repeat Experiments A and B from Section 7.1 with the

[2] For completion, we demonstrate that replacing the Huber loss with our proposed loss function produces comparable results in Appendix E.

[3] The hyper-parameters μ_x, μ_y, μ_w, and μ_h are all set to zero in the implementation, and they are left unchanged in all the experiments.

Table 1: List of Hyper-Parameters

Parameters	Publication		Implementation		Experiment A		Experiment B		Experiment C	
	Proposal	Detection	Proposal	Detection	Proposal	Detection	Proposal	Detection	Proposal	Detection
λ	10	10	1	1	$1/4$	$1/2$	$1/2$	1	$1/4$	1
α	1	1	$1/9$	1	1	1	1	1	1	1
σ_x	1	1	1	$1/10$	$1/20$	$1/20$	$1/20$	$1/20$	$1/20$	$1/20$
σ_y	1	1	1	$1/10$	$1/20$	$1/20$	$1/20$	$1/20$	$1/20$	$1/20$
σ_w	1	1	1	$1/5$	$1/10$	$1/10$	$1/10$	$1/10$	$1/10$	$1/10$
σ_h	1	1	1	$1/5$	$1/10$	$1/10$	$1/10$	$1/10$	$1/10$	$1/10$

Table 2: Interpreted Scale of the Label and Prediction Uncertainties

Bounding Box	Publication		Implementation		Experiment A		Experiment B		Experiment C	
	Proposal	Detection	Proposal	Detection	Proposal	Detection	Proposal	Detection	Proposal	Detection
x^*	w_a	w_a	$w_a/9$	$w_a/10$	$w_a/20$	$w_a/20$	$w_a/20$	$w_a/20$	$w_a/20$	$w_a/20$
y^*	h_a	h_a	$h_a/9$	$h_a/10$	$h_a/20$	$h_a/20$	$h_a/20$	$h_a/20$	$h_a/20$	$h_a/20$
w^*	w_a	w_a	$w_a/9$	$w_a/5$	$w_a/10$	$w_a/10$	$w_a/10$	$w_a/10$	$w_a/10$	$w_a/10$
h^*	h_a	h_a	$h_a/9$	$h_a/5$	$h_a/10$	$h_a/10$	$h_a/10$	$h_a/10$	$h_a/10$	$h_a/10$
$\tilde{x}$	$w_a/10$	$w_a/10$	w_a	$w_a/10$	$w_a/5$	$w_a/10$	$w_a/10$	$w_a/20$	$w_a/5$	$w_a/20$
$\tilde{y}$	$h_a/10$	$h_a/10$	h_a	$h_a/10$	$h_a/5$	$h_a/10$	$h_a/10$	$h_a/20$	$h_a/5$	$h_a/20$
$\tilde{w}$	$w_a/10$	$w_a/10$	w_a	$w_a/5$	$2w_a/5$	$w_a/5$	$w_a/5$	$w_a/10$	$2w_a/5$	$w_a/10$
$\tilde{h}$	$h_a/10$	$h_a/10$	h_a	$h_a/5$	$2h_a/5$	$h_a/5$	$h_a/5$	$h_a/10$	$2h_a/5$	$h_a/10$

Table 3: Faster R-CNN Performance

Parameters	Mean Average Precision (mAP) @		
	0.5 IoU	0.75 IoU	0.5-0.95 IoU
Baseline [18]	41.5	-	21.2
Publication	42.8	18.7	21.0
Implementation	**44.7**	23.1	23.8
Experiment A	**44.7**	24.0	24.2
Experiment B	44.2	**25.0**	24.6
Experiment C	44.6	24.9	**24.7**

Table 4: RetinaNet Performance

Parameters	Mean Average Precision (mAP) @		
	0.5 IoU	0.75 IoU	0.5-0.95 IoU
Implementation	60.1	42.9	40.4
Experiment A	**60.6**	**43.5**	**40.8**
Experiment B	58.9	42.3	39.8

RetinaNet model.[4]

Since RetinaNet is a more recently proposed detector, the implementation supports more sophisticated backbone networks and newer datasets. For all of the experiments, we train the RetinaNet model with a ResNet-101 [8] backbone on the MS-COCO 2017 [15] training set and measure the object detection performance on the validation set utilizing the same metrics as Section 7.1.

The results of the experiments are presented in Table 4. Experiment A was able to achieve higher performance across the board, while Experiment B degraded performance significantly at 0.5 IoU. We observed a similar trend in Section 7.1. These results demonstrate that our proposed interpretation can identify well-suited hyper-parameters for a task regardless of the underlying meta-architecture, backbone network, and dataset.

8. Conclusion

In this work, we propose an alternative probabilistic interpretation of the Huber loss. Our interpretation connects the Huber loss to the KL divergence of Laplace distributions, which provides an intuitive understanding of its parameters. We demonstrated that our interpretation can aid in hyper-parameter selection, and we were able to improve the performance of the Faster R-CNN and RetinaNet object detectors without needing to exhaustively search over hyper-parameters.

The vast majority of recent papers that utilize the Huber loss [1, 2, 3, 6, 7, 10, 11, 13, 16, 22], use the formulation as described in the Fast or Faster R-CNN publications [4, 18]. Therefore, these methods as well as future methods have the potential to be improved significantly by leveraging our proposed interpretation of the Huber loss to identify better suited hyper-parameters for their respective tasks.

[4] Experiment C is not repeated since A and C use the same parameters for the proposal network.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3D object detection for autonomous driving. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[3] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[5] Ross Girshick. Faster R-CNN. `https://github.com/rbgirshick/py-faster-rcnn`, 2015.

[6] Riza Alp Guler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. DenseReg: Fully convolutional dense shape regression in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[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 (CVPR)*, 2016.

[9] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 03 1964.

[10] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3D proposal generation and object detection from view aggregation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[11] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3D object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] Kenneth Lange. Convergence of EM image reconstruction algorithms with Gibbs smoothing. *IEEE Transactions on Medical Imaging*, 9(4):439–446, 1990.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[17] Asaf Noy and Koby Crammer. Robust forward algorithms via PAC-Bayes and Laplace distributions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[20] Guijin Wang, Xinghao Chen, Hengkai Guo, and Cairong Zhang. Region ensemble network: Towards good practices for deep 3D hand pose estimation. *Journal of Visual Communication and Image Representation*, 55:404–414, 2018.

[21] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[22] Bin Yang, Wenjie Luo, and Raquel Urtasun. PIXOR: Real-time 3D object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.