Shweta Kanhere(9049988461)

Email: shweta.kanhere@gmail.com

Huber Loss

The study of Outliers is important to know the variation in the data

In some fields to know the change in trends is represented by outliers like to see the reaction action of any drugs, to see the changes in trends of any consumable.

So please be aware of where to delete/treat/observe the outliers.

Generally, the MSE is great for learning outliers while the MAE is great for ignoring them. When the data has outliers then we use MAE as it deals with absolute value and if the data has no outliers then we have to use MSE as its deals with the square of error.

But if we want something in the middle then what technique do we need to use?

Consider an example where we have a dataset of 100 values we would like our model to be trained to predict. Out of all that data, 25% of the expected values are 5 while the other 75% are 10.

An MSE loss wouldn't quite do the trick, since we don't really have "outliers"; 25% is by no means a small fraction. On the other hand we don't necessarily want to weight that 25% too low with an MAE. Those values of 5 aren't close to the median (10 — since 75% of the points have a value of 10), but they're also not really outliers.

Solution for above problem?.......... The Huber Loss Function.

It is the combination of MAE and MSE. So its best solution.

The Huber Loss offers the best of both worlds by balancing the MSE and MAE together. We can define it using the following piecewise function:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\, |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

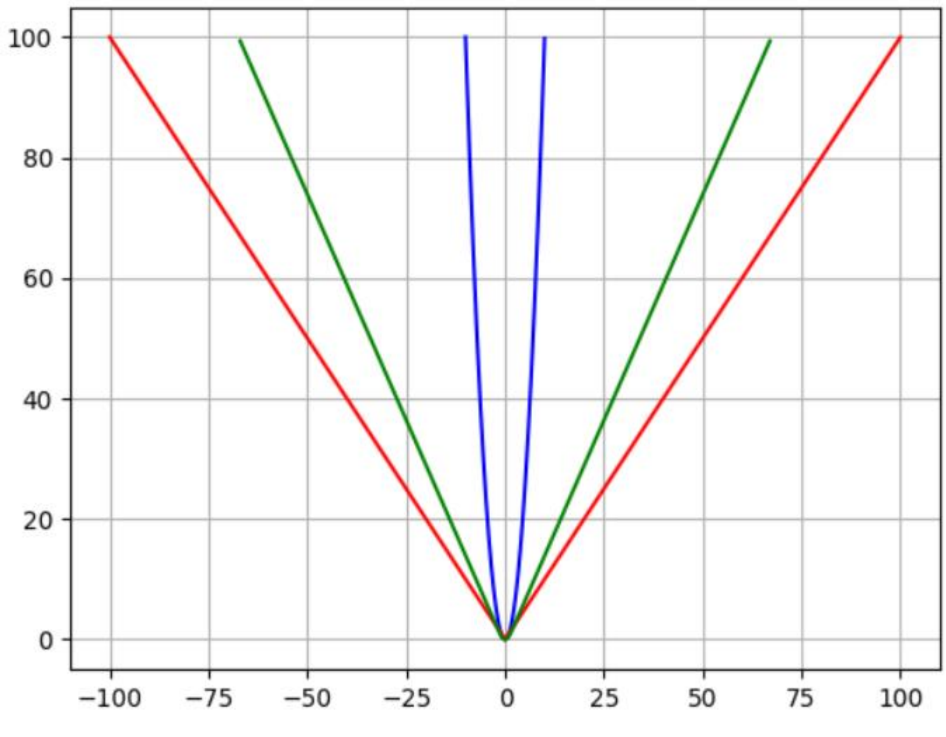
What this equation essentially says is: for loss values less than delta, use the MSE; for loss values greater than delta, use the MAE. This effectively combines the best of both worlds from the two loss functions!

If delta =1 then loss function = y – 0.5

Using the MAE for larger loss values mitigates the weight that we put on outliers so that we still get a well-rounded model. At the same time we use the MSE for the smaller loss values to maintain a quadratic function near the centre.

This has the effect of magnifying the loss values as long as they are greater than 1. Once the loss for those data points dips below 1, the quadratic function down-weights them to focus the training on the higher-error data points.

Check out the code below for the Huber Loss Function. We also plot the Huber Loss beside the MSE and MAE to compare the difference.



MAE (red), MSE (blue), and Huber (green) loss functions

Image Reference: https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3

Code :

```
import matplotlib.pyplot as plt

import numpy as np


# Huber loss function

def huber_loss(y_pred, y, delta=1.0):

    MSE_huber= 0.5*(y-y_pred)**2

    MAE_huber= delta * (np.abs(y - y_pred) - 0.5 * delta)

    return np.where(np.abs(y - y_pred) <= delta, MSE_huber, MAE_huber)


# Plotting

x_vals = np.arange(-65, 65, 0.01)

delta = 1.5

MSE_huber= 0.5*np.square(x_vals)

MAE_huber= delta * (np.abs(x_vals) - 0.5 * delta)

y_vals = np.where(np.abs(x_vals) <= delta,MSE_huber, MAE_huber)

plt.plot(x_vals, y_vals, "green")

plt.grid(True, which="major")

plt.show()
```