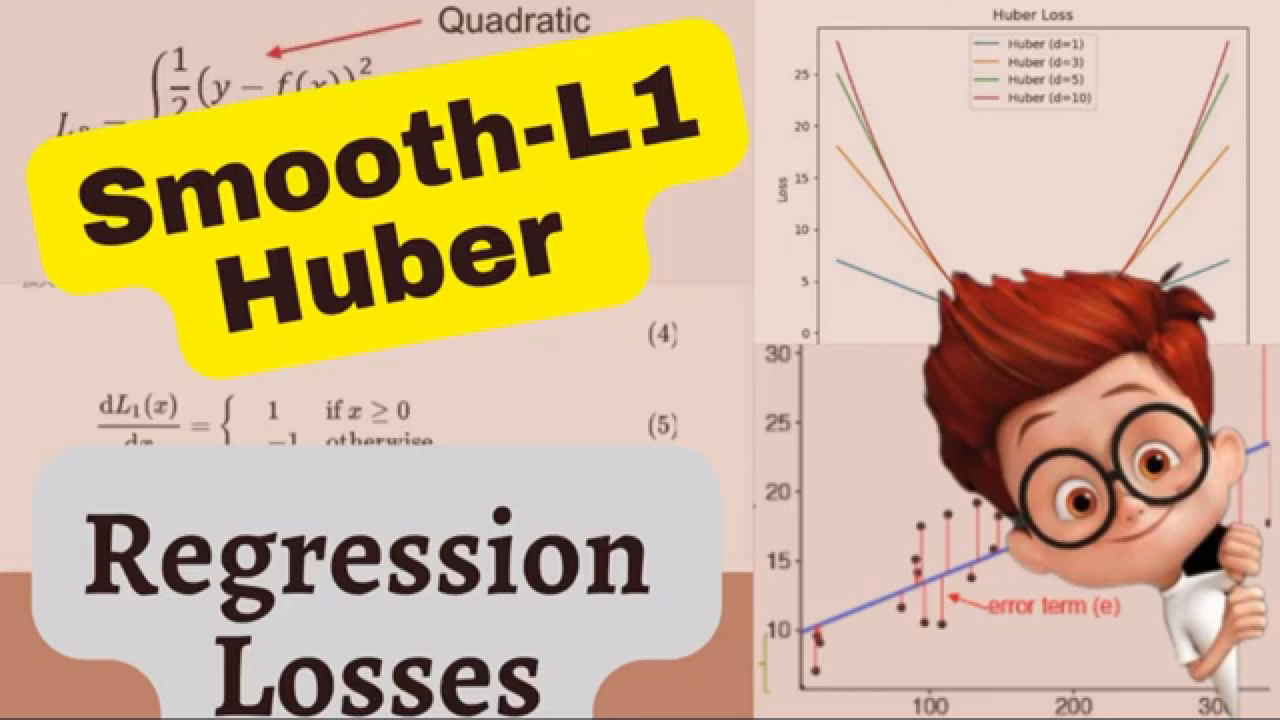
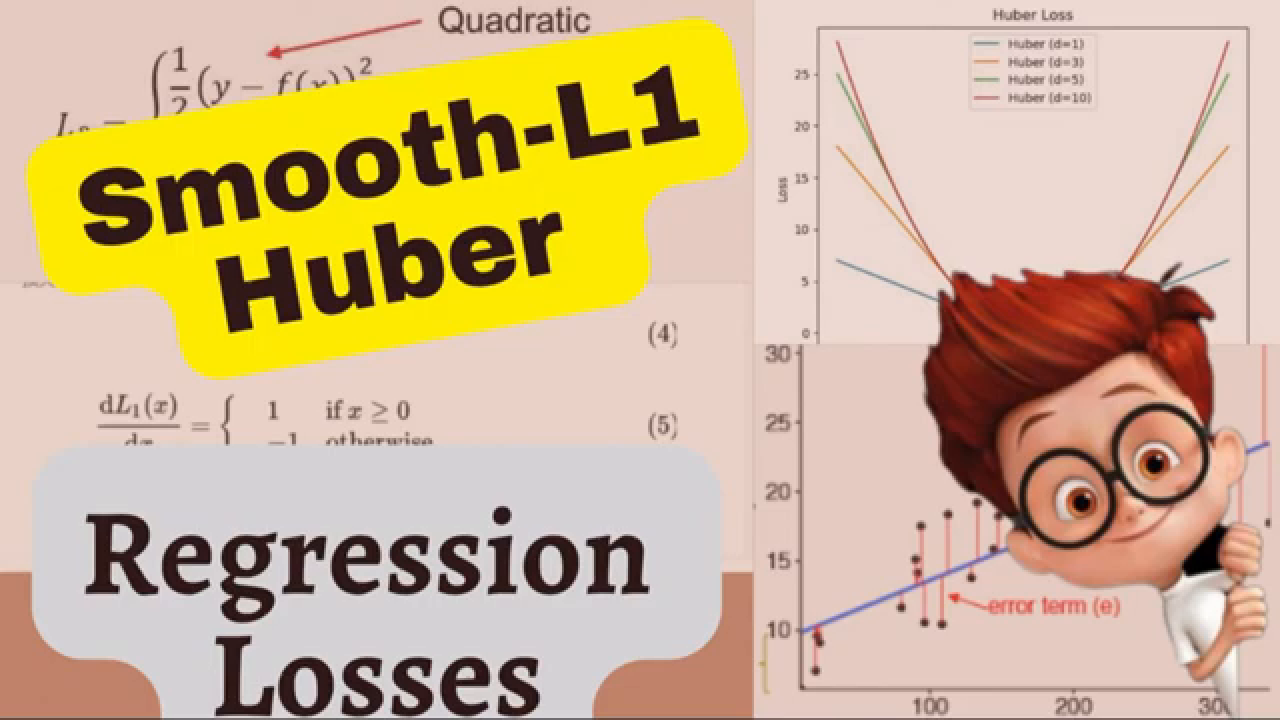
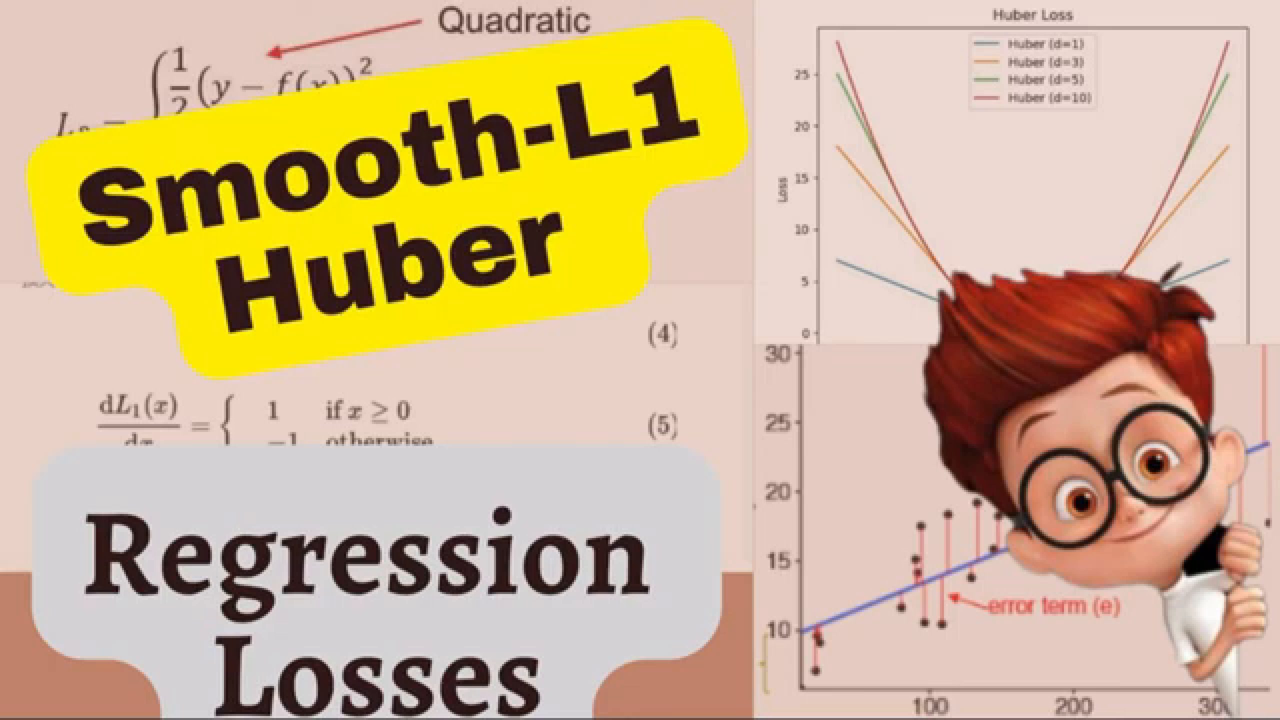


Quadratic
Smooth-L1
Huber
(4)
(5)
Regression
Losses
Huber Loss
Huber (d=1)
Huber (d=3)
Huber (d=5)
Huber (d=10)
Loss
error term (e)

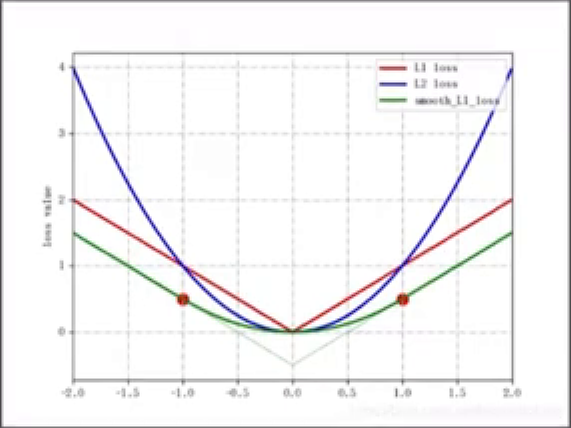

Quadratic
Smooth-L1
Huber
(4)
$\frac{dL_1(x)}{dx} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$
(5)
Regression
Losses
Huber Loss
Huber (d=1)
Huber (d=3)
Huber (d=5)
Huber (d=10)
Loss
error term (e)


Quadratic
Smooth-L1
Huber
(4)
(5)
Regression
Losses
Huber Loss
Huber (d=1)
Huber (d=3)
Huber (d=5)
Huber (d=10)
Loss
error term (e)

Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

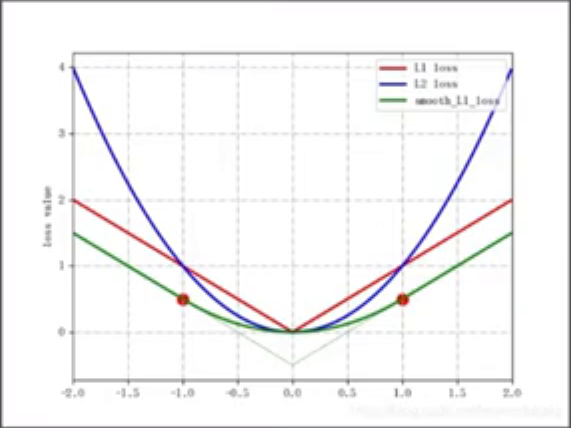
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

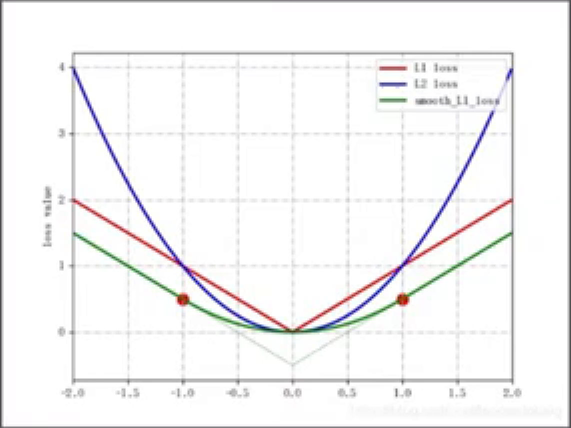
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

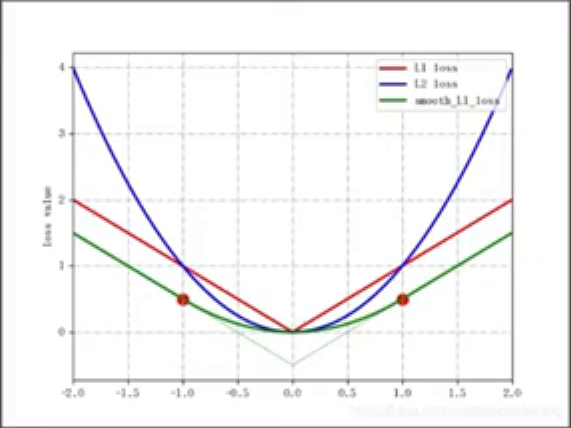
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

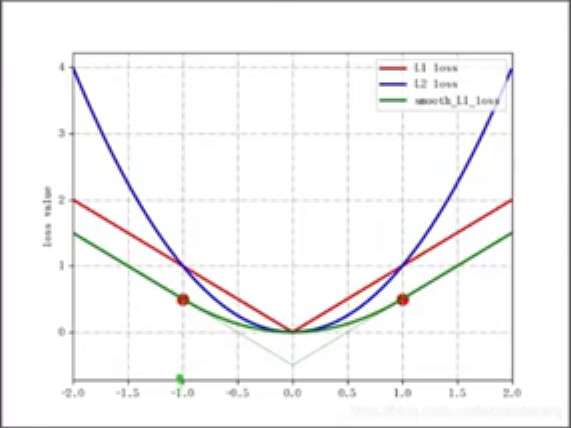
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

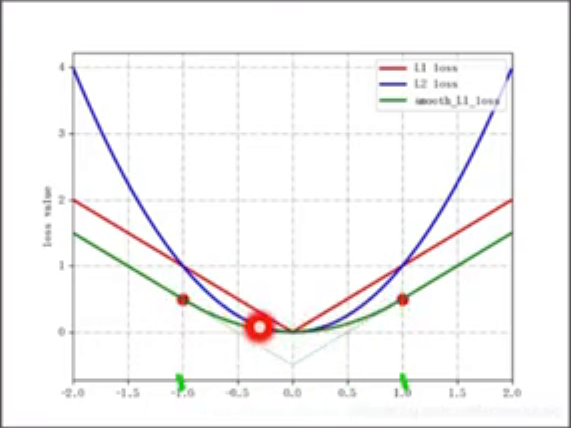
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

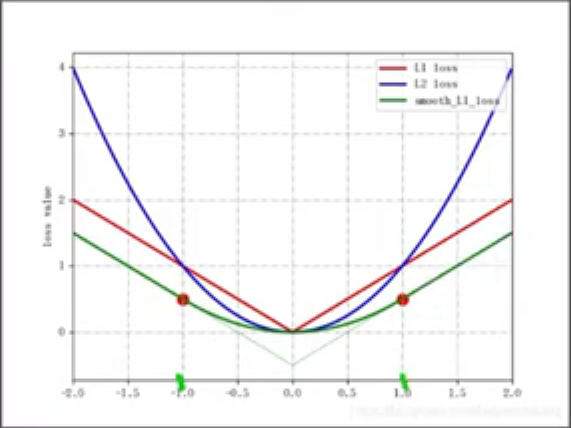
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

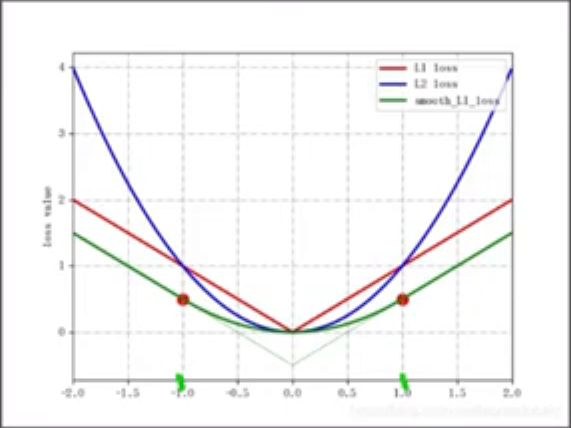
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

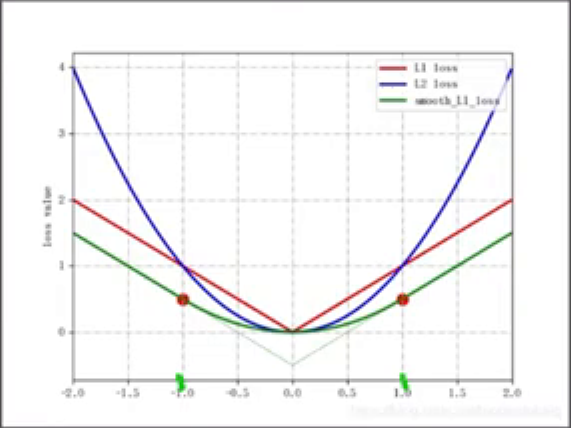
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

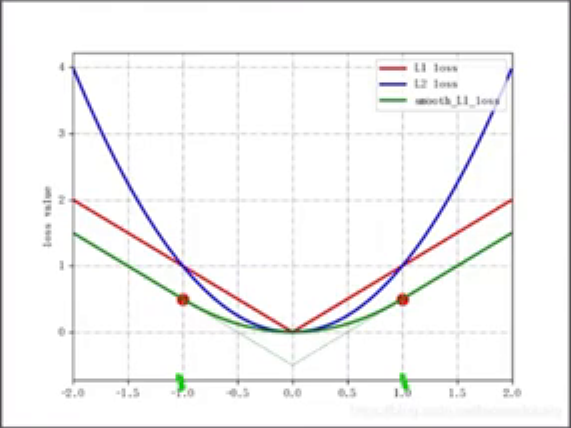
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

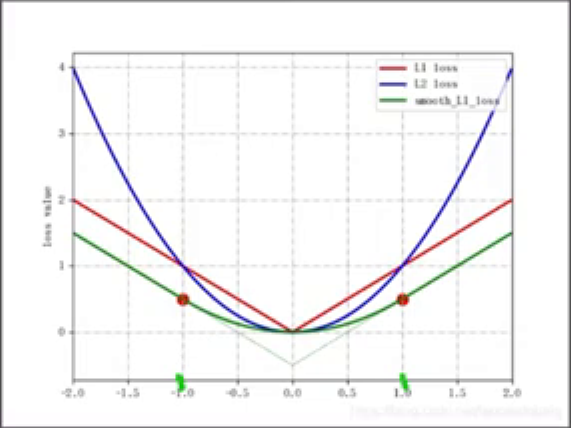
where x = (Y^-Y)
Y^ - Prediction
Y - Target



Smooth L1 loss

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

where x = (Y^-Y)
Y^ - Prediction
Y - Target



Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Advantages

- Advantages of both L1 loss and L2 loss
- Less sensitive to outliers than L2 loss
- Prevents Exploding Gradients
- Well suited for most regression tasks

Derivative

$$L_2(x) = x^2$$

$$L_1(x) = |x|$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Derivative

$$L_2(x) = x^2$$

$$L_1(x) = |x|$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}L_2(x)}{\mathrm{d}x} = 2x$$

$$\frac{\mathrm{d}L_1(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Derivative

$$L_2(x) = x^2$$

$$L_1(x) = |x|$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}L_2(x)}{\mathrm{d}x} = 2x$$

$$\frac{\mathrm{d}L_1(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}\ \text{smooth}_{L_1}}{\mathrm{d}x} = \begin{cases} x & \text{if } |x| < 1 \\ \pm 1 & \text{otherwise} \end{cases}$$

Derivative

$$L_2(x) = x^2$$

$$L_1(x) = |x|$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}L_2(x)}{\mathrm{d}x} = 2x$$

$$\frac{\mathrm{d}L_1(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}\,\text{smooth}_{L_1}}{\mathrm{d}x} = \begin{cases} x & \text{if } |x| < 1 \\ \pm 1 & \text{otherwise} \end{cases}$$

Derivative

$$L_2(x) = x^2$$

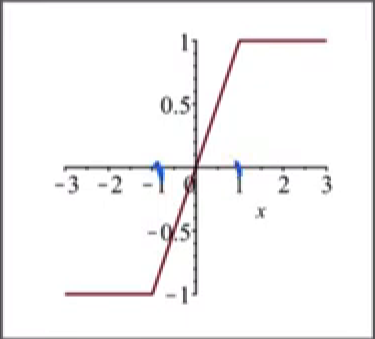
$$L_1(x) = |x|$$

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}L_2(x)}{\mathrm{d}x} = 2x$$

$$\frac{\mathrm{d}L_1(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}\ \mathrm{smooth}_{L_1}}{\mathrm{d}x} = \begin{cases} x & \text{if } |x| < 1 \\ \pm 1 & \text{otherwise} \end{cases}$$



Derivative

$$L_2(x) = x^2$$

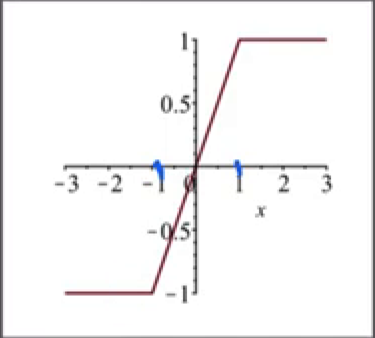
$$L_1(x) = |x|$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}L_2(x)}{\mathrm{d}x} = 2x$$

$$\frac{\mathrm{d}L_1(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}\ \text{smooth}_{L_1}}{\mathrm{d}x} = \begin{cases} x & \text{if } |x| < 1 \\ \pm 1 & \text{otherwise} \end{cases}$$



Derivative

$$L_2(x) = x^2$$

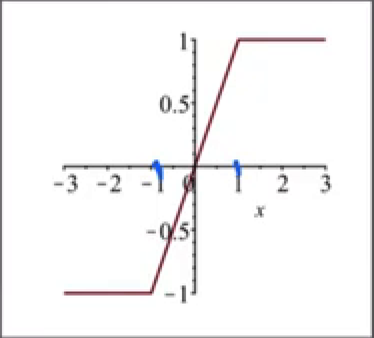
$$L_1(x) = |x|$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}L_2(x)}{\mathrm{d}x} = 2x$$

$$\frac{\mathrm{d}L_1(x)}{\mathrm{d}x} = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

$$\frac{\mathrm{d}\ \text{smooth}_{L_1}}{\mathrm{d}x} = \begin{cases} x & \text{if } |x| < 1 \\ \pm 1 & \text{otherwise} \end{cases}$$



Any Problem?

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Any Problem?

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Any Problem?

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Why 1?

Can I use 2 or 5?

Any Problem?

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Why 1?

Can I use 2 or 5?

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$\Downarrow$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Parameterized

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Parameterized

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Parameterized

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Parameterized

Any issues?

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise}, \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Any issues?

Parameterized

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise}, \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Any issues?

Parameterized

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Any issues?

Parameterized

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise}, \end{cases}$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

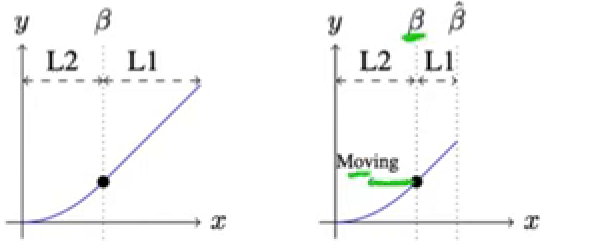
Parameterized

Any issues?

Dynamic SmoothL1 Loss

Dynamic SmoothL1 Loss

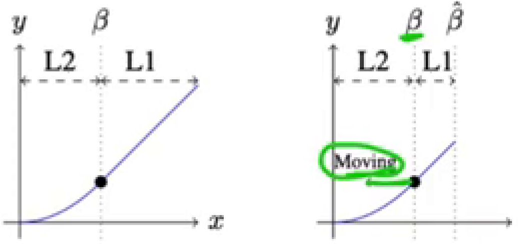
Self-Adjusting Smooth L1 Loss



(a) Smooth L1 (b) Self-Adjusting Smooth L1

Dynamic SmoothL1 Loss

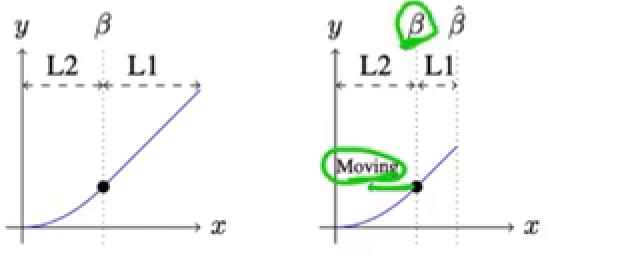
Self-Adjusting Smooth L1 Loss



(a) Smooth L1 (b) Self-Adjusting Smooth L1

Dynamic SmoothL1 Loss

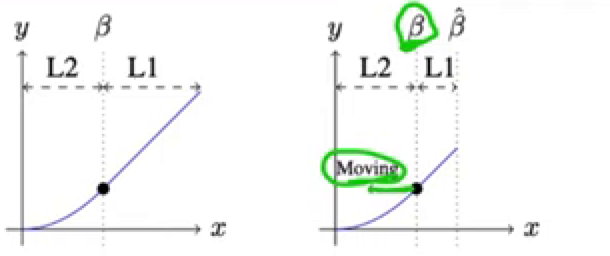
Self-Adjusting Smooth L1 Loss



(a) Smooth L1 (b) Self-Adjusting Smooth L1

Dynamic SmoothL1 Loss

Self-Adjusting Smooth L1 Loss



(a) Smooth L1 (b) Self-Adjusting Smooth L1

Huber Loss

https://github.com › torch › issues

Smooth L1 loss vs Huber loss · Issue #579 · torch/nn - GitHub

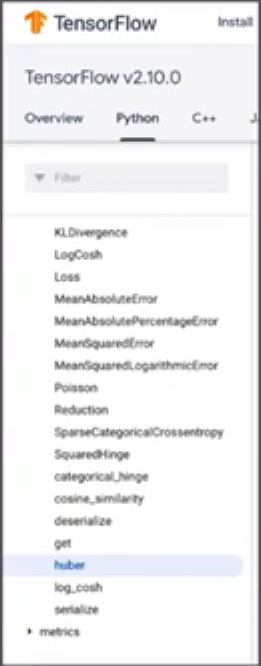
22-Jan-2016 — I would say that **the Huber loss really is parameterised by delta**, as it defines the boundary between the squared and absolute costs. I don't ...

Huber Loss

https://github.com › torch › issues

Smooth L1 loss vs Huber loss · Issue #579 · torch/nn - GitHub

22-Jan-2016 — I would say that **the Huber loss really is parameterised by delta**, as it defines the boundary between the squared and absolute costs. I don't ...



Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

beta = delta

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber = beta (S-L1)

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } \underline{|x_n - y_n| < delta} \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber = beta (S-L1)

Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2 / beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

Smooth-L1

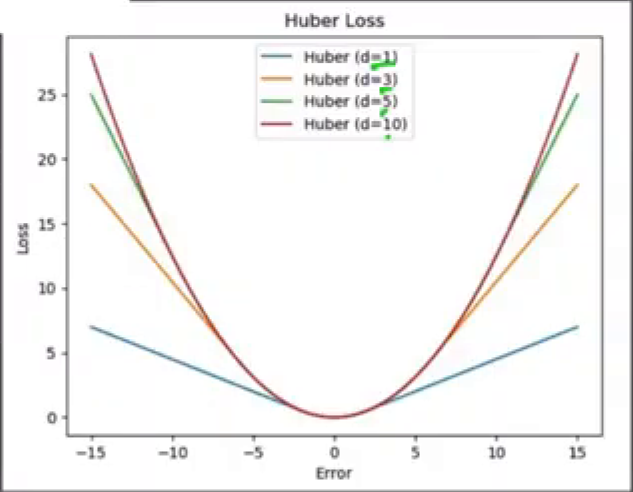
$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Huber = beta (S-L1)

3

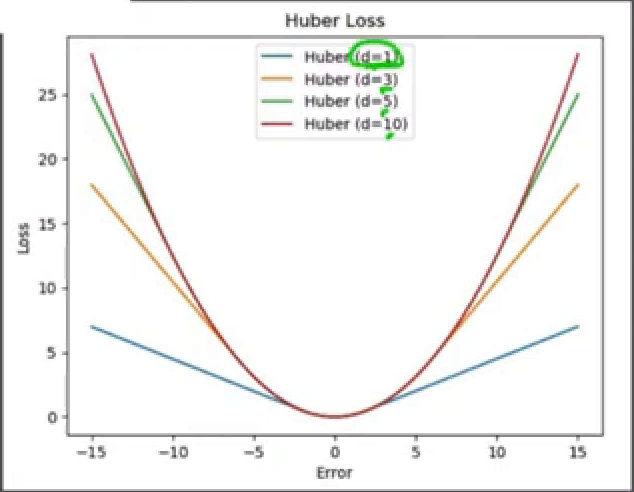
Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$



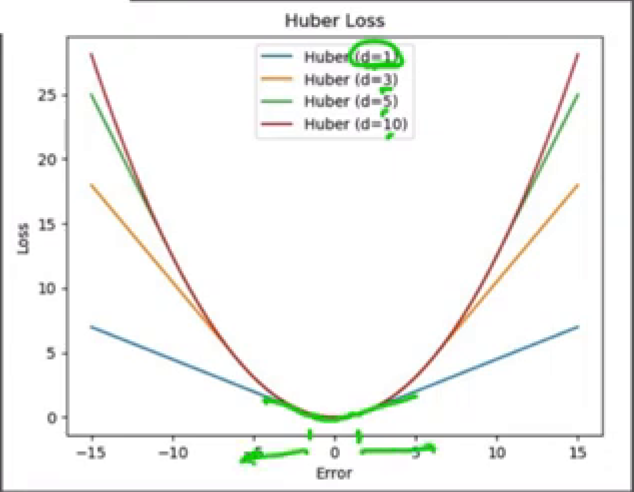
Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$



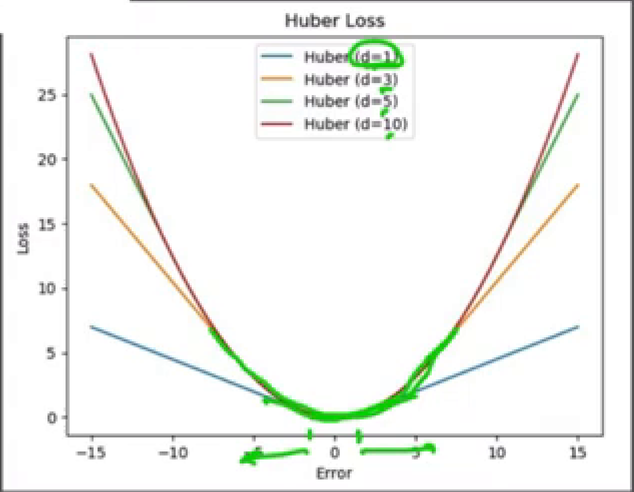
Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$



Huber Loss

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$



Implementation

```
def smoothl1loss(x, y):
    if abs(x-y)<1: return 1/2*(x-y)**2
    else: return abs(x-y)-1/2
```