Today's Agenda

1) Loss Functions
2) Activation Functions

Loss
||
Error

Loss Func / Error {Cost Function}

What ?

Method of evaluating how well our algorithm is working on the dataset.

$= (y - \hat{y})^2$

Loss → Single Data Point | Cost → Batch/Multiple data

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Loss Function	Cost Function
Measures the error between predicted and actual values in a machine learning model.	Quantifies the overall cost or error of the model on the entire training set.
Used to optimize the model during training.	Used to guide the optimization process by minimizing the cost or error.
Can be specific to individual samples.	Aggregates the loss values over the entire training set.
Examples include mean squared error (MSE), mean absolute error (MAE), and binary cross-entropy.	Often the average or sum of individual loss values in the training set.
Used to evaluate model performance.	Used to determine the direction and magnitude of parameter updates during optimization.
Different loss functions can be used for different tasks or problem domains.	Typically derived from the loss function, but can include additional regularization terms or other considerations.

Loss Functions → Any one loss function

Cost Function → Anyone OR Combination of multiple loss functions

Object Detection → Regression (Location) + Classification (Class) D/C

→ loss Function

X Y Z Y

Once the loss is calculated

↓

(Min) update weights / biases (B.P)

Cost Function = L.F + Regularization

Loss Functions

1) Regression

1) MSE
2) MAE
3) Huber Loss

2) Classification

1) Binary C·E
2) Categorical C·E
3) Sparse Categorical C·E

Auto Encoders

i) KL Divergence

GAN

i) Discriminator Loss
ii) min max Gan loss

Object Detection

i) Focal Loss

Word Embeddings

i) Triplet Loss

Mean Squared Error (L2 Loss / Squared Loss)

$$\text{Loss} \rightarrow (y_i - \hat{y}_i)^2$$

$$\text{Cost} \rightarrow \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

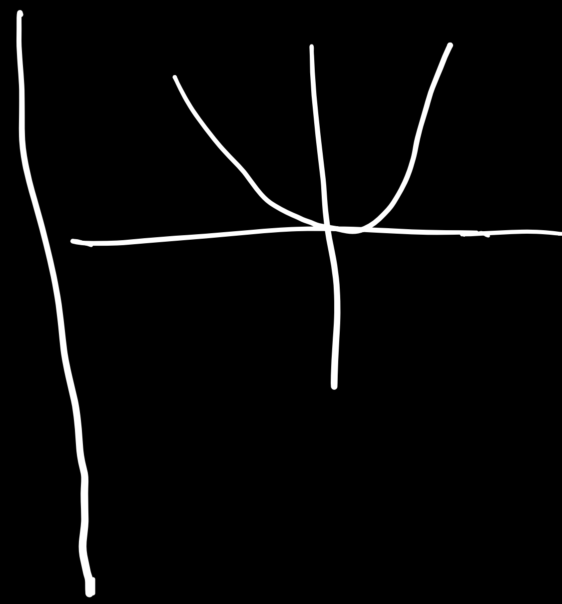
Pros

1) Ease to Interpret

2) Differentiable

3) Single Minima

→ Smooth Convergence



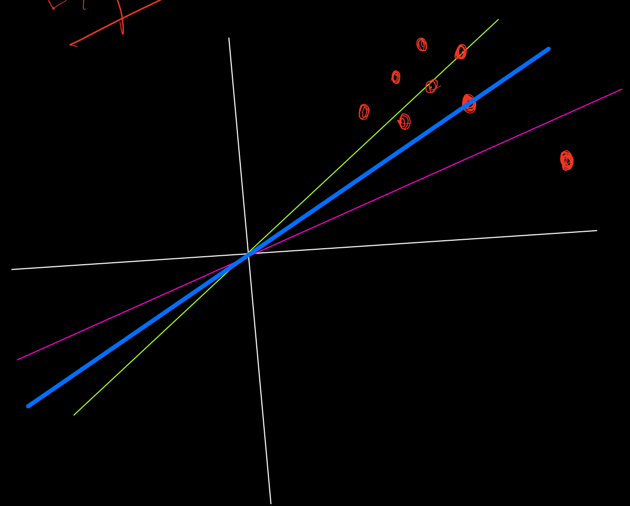
CGPA	12^{th} Mark	Y (LPA)	$\hat{Y}$
8.1	75	6.1	5.8

Big W/B update

Exploding Gradients

Soln = $\sqrt{\ }$

$$\text{Loss} = (\text{Actual} - \text{Pred})$$
$$= (6.1 - 5.8)$$
$$= (.3)^2$$
$$= .09$$



Magnitude of Error

Closer → less

Far → More

Units of Error

1 unit = $(1)^2$ Error

2 units = $(2)^2$ Error

4 units = $(4)^2$ Error

Outlier

1) Prone (MSE)

Cons of MSE

1) Squared Error Unit
2) Outlier Issue

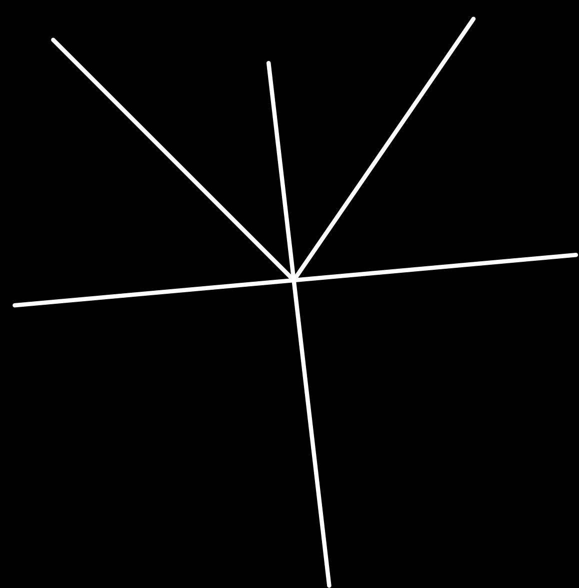
Mean Absolute Error (L1 loss)

$$L \cdot F = |Y_i - Y_i|$$

$$C \cdot F = \frac{1}{n} \sum_{i=1}^{n} |Y_i - Y_i|$$

Pros

1) Easy to understand
2) Error Unit Same
3) Robust to Outliers



Cons

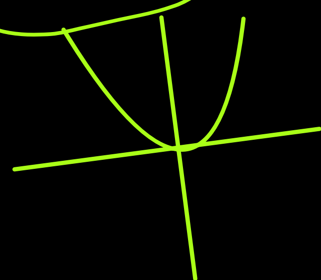
1) Not Differentiable → Subgradient
2) Time Complexity high

Huber loss

Combination of {MSE + MAE}

$$\text{Huber} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| \le \delta \\ \frac{1}{n}\sum_{i=1}^{n} \delta\, |y_i - \hat{y}_i| - \frac{1}{2}\delta \end{cases}$$

δ → hyperparameter



Pros

1) Differentiable
2) Very Robust to Outlier
3) Smooth Convergence

Con

1) Hyperparameter δ (optimizing its value)
→ Calculation is time high

25% data are outliers → Huber

2% data are outliers → MAE

0% data → MSE

Classification

B.C.E (Binary Cross Entropy) / Log loss

$$L.F = -y \log \hat{y}_i - (1-y_i)\log(1-\hat{y}_i)$$

1) Binary Classification (2 classes)

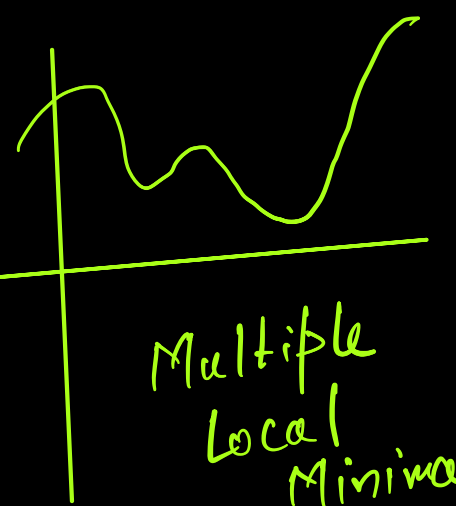
$$C.F = -\frac{1}{n}\sum_{i=1}^{N} y \log \hat{y}_i - (1-y_i)\log(1-\hat{y}_i)$$

Pros

1) Differentiable

Cons

1) Rough Convergence
2) Multiple local Minima
3) Hard to interpret



Sigmoid

CGPA	12th Mark	Y|N	$\hat{Y}$
7.1	75	1	0.73
6.1	63	0	N

$$= -1 \log(0.73)$$
$$= -1 \times (-0.13) = \underline{\underline{0.13}}$$
$$= -1$$

Categorical Cross Entropy

Multiclass > 2 classes, no of classes

$$L.F = -\sum_{j=1}^{K} Y_j \log(\hat{Y}_j)$$

K = 3

$$L.F = -Y_1 \log(\hat{Y}_1) - Y_2 \log(\hat{Y}_2) - Y_3 \log(\hat{Y}_3)$$

Softmax

$$= \frac{e^{z}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

CGPA	12marks	Placed
8	80	Yes
6	60	No
7	70	Maybe

Categorical

$$\begin{array}{cccc} & \text{Yes} & \text{No} & \text{Maybe} \\ \text{Yes} & [1 & 0 & 0] \\ \text{No} & [0 & 1 & 0] \\ \text{Maybe} & [0 & 0 & 1] \end{array}$$

$$= [0.2, 0.3, 0.5]$$

$$=$$

$$= -1 \cdot \log(0.2)$$

$= 0.69$

$$\text{Cost Function} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{j}\left[-y_{ij}\log(\hat{y}_{ij})\right]$$

Pros

1) Simplest loss function for M·C·C

Cons

1) Time Taking
2) One Hot Encoding

Sparse Categorical Cross Entropy

L·F & C·F are Same to **CCE**

Labels

CGPA	12th	Mark	
8	80	Yes	1
6	60	No	2
7	70	Maybe	3

C·C·E | **Difference** | **S·C·C·E**

C·C·E:
1) Output (one-hot Encoded)

S·C·C·E:
1) Integers