Loss function

↓

Method of evaluating how well your model is being trained on the provided dataset.

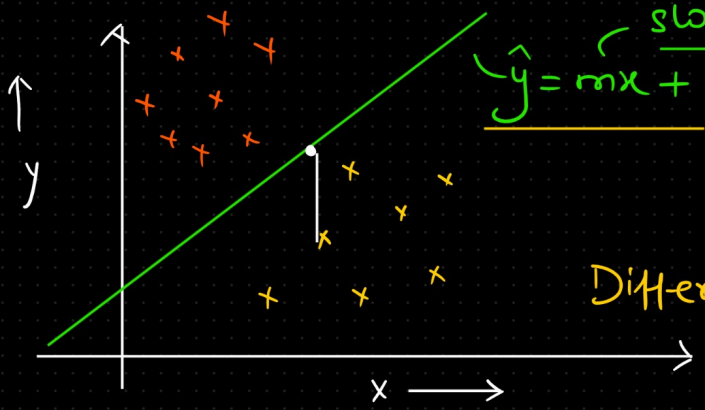
Loss function value

- High → Poor
- low → Really great job

Ideal Case

ML Algorithms ← Linear Regression (find the best fit line)

↳ (Regression Tasks)



$\hat{y} = mx + c$

slope

intercept

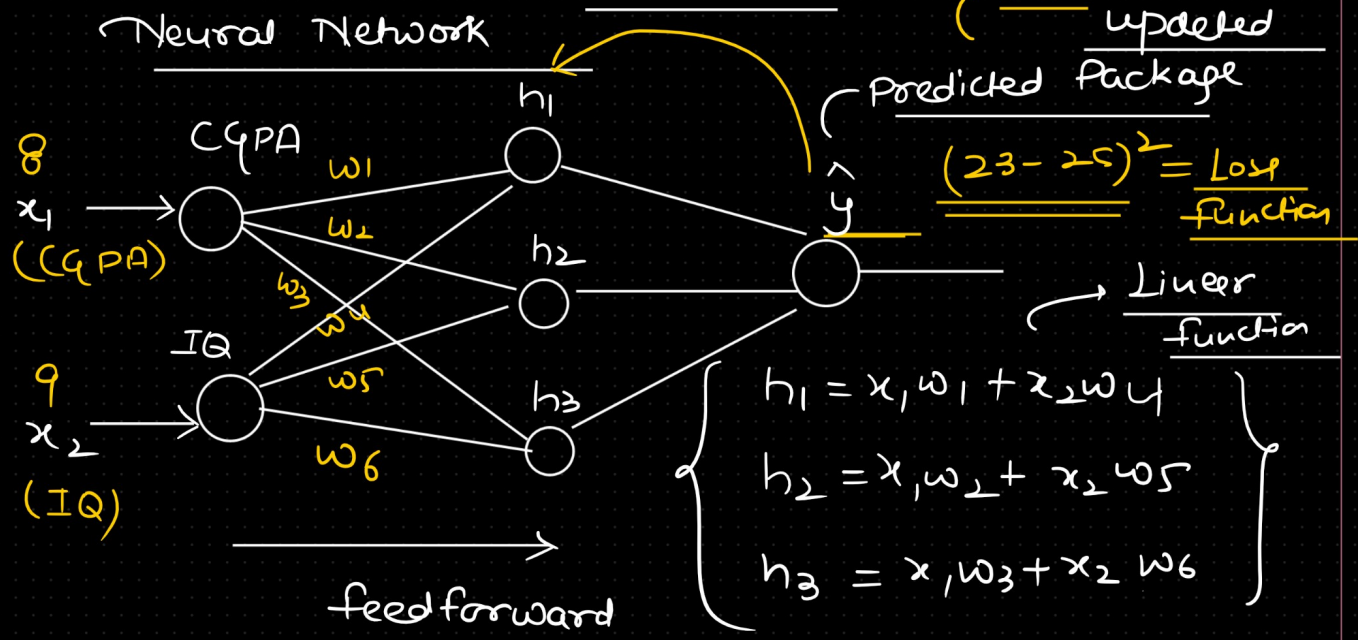
Difference b/w Loss-function vs Cost function

- Loss function $\Rightarrow (y_i - \hat{y}_i)^2$ → single training data
  - Actual
  - Predicted
  - MSE
- Cost function $\Rightarrow \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ → batch

Deep Learning

CGPA	IQ	Package (LPA) y ← target	Predicted $\hat{y}$
8	9	25	23
6	7	3.4	3.2
5	6	2.5	2.7
7	9	12	11
9	9	80 ✓	30 ✓

Outliers → backpropagation → weights & biases will be updated



Training → Minimum value of Loss function

Peter Drucker

→ You can't improve what you can't measure

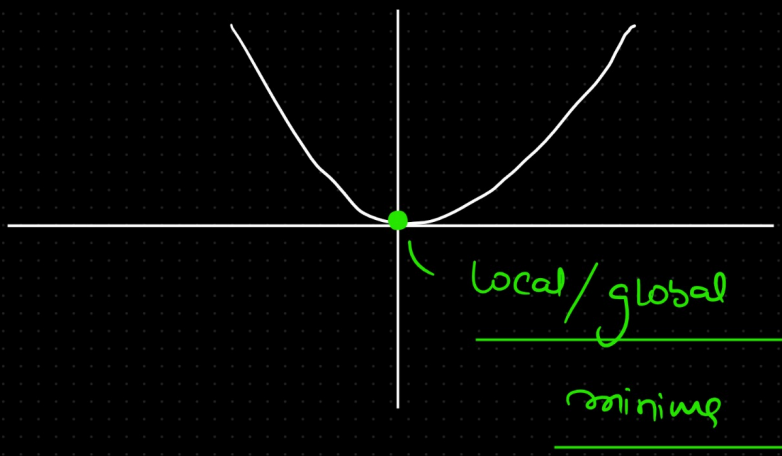
Motivation

Different types of loss function

Regression → Prediction of any continous value

1. MSE → Mean Squered Error
2. MAE → Mean Absolute Error
3. Huber Loss

MSE
1. Easy to interpret
2. Differentiable
3. Only one local minima



Disadvantage → Not robust to outliers

MAE $\longrightarrow$ $|y_i - \hat{y}_i| \Rightarrow$ Loss function

$\frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \Rightarrow$ Cost function

- Outliers $\longrightarrow$ More robust to outliers

Disadvantage

↳ Not differentiable

Huber loss $\rightarrow$ Combo of MSE & MAE

↳ Robust to outliers

Classification → ① Binary Cross entropy

ML (Logistic regression) → Log loss

↳ Binary Classification tasks (sigmoid activation function)

$$-y \log(\hat{y}) - (1-y) \log(1-\hat{y}) = \text{loss function}$$

② Categorical cross entropy

↳ multiclass classification tasks → Softmax activation function

$$L = \sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

③ Hinge loss → SVM (ML)

Autoencoders

↳ KL Divergence

GAN (generate synthetic data)

① Discriminator loss

② Minmax GAN loss

Embedding → Triplet Loss