Loss Function

Backpropogation Mathmatical Approch -

Training → 1) Feed Forward propogation
2) Loss function
3) Back ward propogation

① Loss function →

Def ⇒ It is a Method for evaluting the algo its performance ⇒

x	y	o
1	5	3
5	8	7
6	1	8

o → y = Actual output

$\hat{y}$ = Predicted output

$\hat{y}$
3.6
6.6
4

Error ⇒

Error = Actual − predicted

Neural = Minus → Error ⇒ Mathematical

Reason of Multiple loss function →

1 →
2 →
3
4 →

4) Use different loss function
1) Use Cass
2) Neural Network
3) Activation →

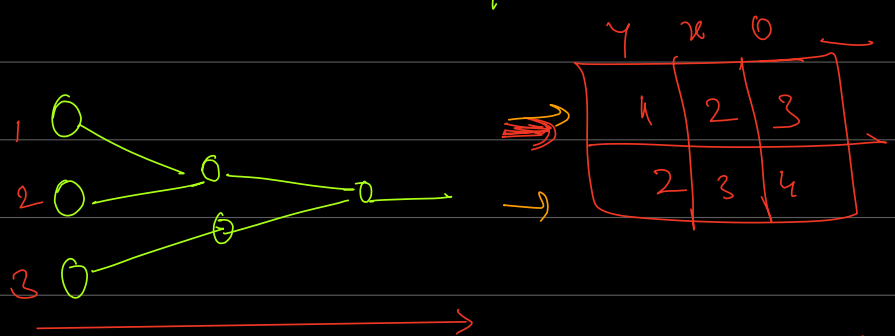
→ different Mathematical function

loss function

(2) Types of loss function

(3) * Cost function & * loss function →

Single Iteration



1) Forward propogation
2) loss function

Error

A-Error

loss function

Cost function

Normal Error

Average Error

MSE → $(y_i - \hat{y})^2$

Actual predicted

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

Types of loss function

- Regression
  1) MSE (Mean square Error)
  2) MAE = (Mean Absolute Error)
  3) Huber loss
- Classification
  1) Binary Cross Entropy
  2) Categorical Cross E-
  3) sparse C. Cross Entro

Dataset

Independent	dependent
Input	output

dependent = output

① Numerical = Regression

② Categorical = classification

A	B	C	O/
1	40	70	1
2	10	60	3

A	B	C	O	
1	60	20	Yes	1
2	50	60	No	0

3	80	50	5

3	70	32	Yes	0

Regression → Numerical

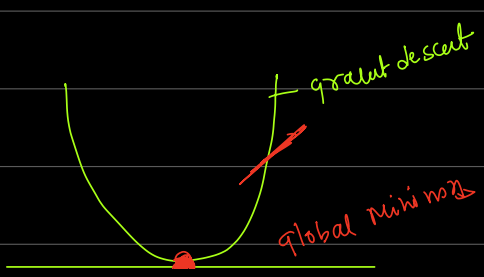
Category

→ Regression → Loss function

- Formula
- When to use
- Advantage
- Disadvantage
- Activation function

Layers → { Hidden, Output }

① What is global & local Mining ?



$i = 1 \to 6$

Global

Loss → Convergence = Minima Error

Global point where error = minima

(1) (2) equation of loss function = Quadric = global minima

(1) MSE :-

Mean Squared Error ⇒ 1) MSE (Mean Square Error)

2) Squared loss

3) l^2 loss

(2).

$$MSE = \frac{1}{N}\sum_{i}^{N}(Yi - \hat{Y}i)^2$$

No of value Actual predicted

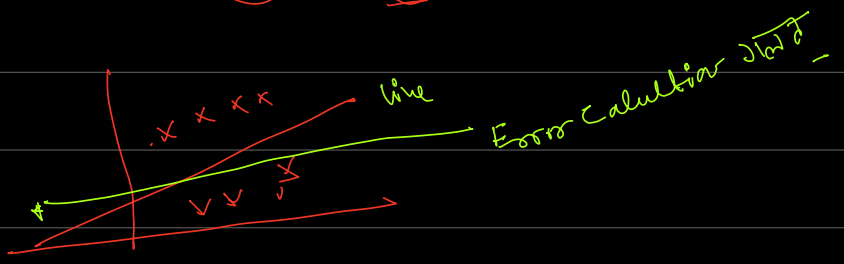
(2) → where to use → 1) Regression

2) No outlier present

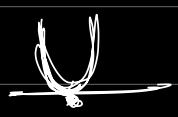
$(0.1)^2$

(0.9)

x	y	0.1	0.4
1	(5) → (25)		expand
6	(8) → (64)		



(2). Differential easily ⇒ x^2 ⇒ differentiable



Global minima

Advantages ⇒ (1) Easy interpretation

(2) → Always Differentiable

(3) Global minima.

Disadvantages : 1) Not Robust to outliers

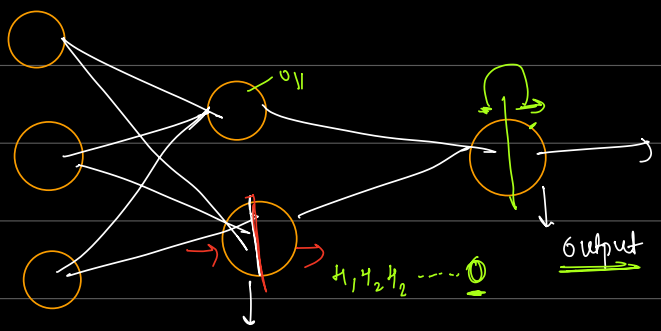
↓

(Not able to Handle)

(2). Sensitive to overfitting.

Activation function :- Hidden layer : ReLU

output layer : Linear Activation fun'n.



Hidden Layer

ReLU → to add Non-linearity

to learn complex pattern

Output linear → we use linear because we want the output

2. MAE

1) Mean Absolute Error
2) LL loss function

Cost F ⇒

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |Y_i - \hat{Y}_i|$$

Y_i ↓ Actual, $\hat{Y}_i$ ↘ predicted

Difference ⇒ $|y - \hat{y}|$ → Square Removed by mode

Why ? Mode ~ ( ) → x

→ y $\hat{y}$

3	2	⇒ 1 my
1	4	-3 map
5	2	3 my

← Mode

Actual Error

$|1 - 3| \rightarrow (3)$

$1 + 3 + 3 \rightarrow (7)$

⇒ 1, 3, 3

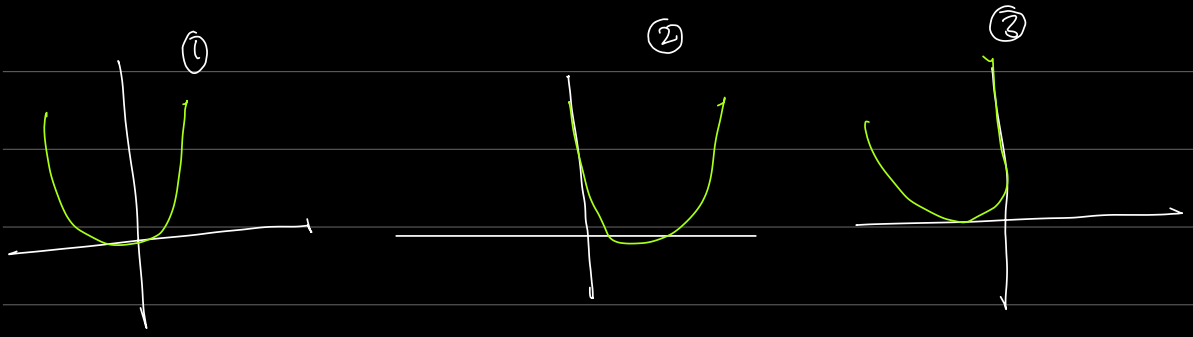
(*)

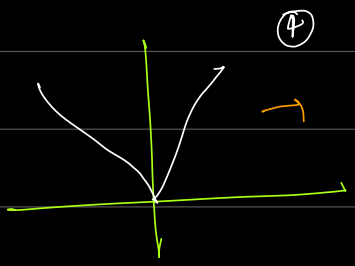
Where to use → 1) Dataset containing Outlier

2) Regression

(2) Regression

(3) It is not differentiable





→ Median → distribution

→

Advantages → 1) Robust to ~~outlier~~

2) Easy to interpretate

Disadvantage → 1) It is not differentiable

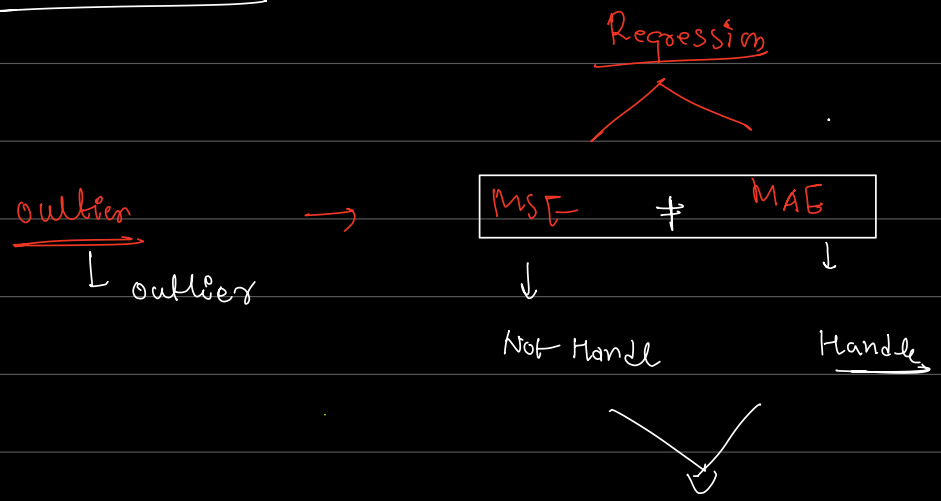
2) Less sensitive to large error

(3) → Not able find the direction of graph

Activation function → Hidden → ReLu (Non linearity)

Output ⇒ Linear

③ Huber loss ⇒ Reason : ⇒



1) global
2) differential
3) outlie

↳ Huberloss

1) MSE = ∪ global

+

3) MAE → Outlier Handle

Not

When to prefer which ⇒

X → 1) Large dataset →

X → 2) Slow processing (It is slow)

Formula ⇒

$$Huber = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(y_i - \hat{y}_i)^2 \qquad |y_i - \hat{y}_i| \le \delta \quad \text{if} \le \delta$$

$$Huber = \frac{1}{n}\sum_{i=1}^{n}\delta\left(|y_i - \hat{y}_i| - \frac{1}{2}\delta\right) \qquad |y_i - \hat{y}_i| > \delta \quad \text{if} > \delta$$

n = number of data points

y = Actual value

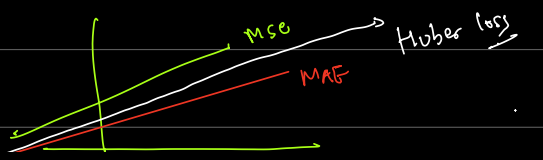
$\hat{y}$ = predicted value.

δ = Define point where Hubers loss transition from Quadratic to linear

Condition = threshold

(1) Advantages ⇒ 1) It handle the outlier dataset as well

(2) - It lies b/w MAE and MSE



(2) → 9t is differentiable → global minima

Disadvantages →

1) 9t is slow processing.
2) 9t is not good large dataset
3) 9t is complex
4) 9t need hyperparter &

Activation Function → Hidden layer → Relu

output → Linear AF

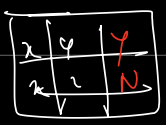
Classification

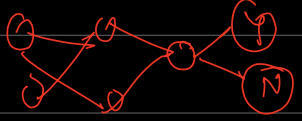
1) Binary
2) Multi calass
3) spars

(1) Binary Cross Entropy ⇒ (1) **Log loss** fonction

Where → 1) Classification

2) Binary classification





(8) → Cost –

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

Loss – $-y \log(\hat{y}) - (1-y)\log(1-\hat{y})$

Actual — Predicted

Why log ? → (1) Probabilistic Answers

(2) Penality to incorrect predictions

(3) Smoother Gradient

(*) Advantages → 1) It penalize incorrect predictions

2) Differentiable

3) For probabilistic interpretation

* Disadvantage ⇒ 1) problem of local minima
2) Not intuitive
3) It depend on class imbalance
(sensitive to class imbalance)

→ Dataset input ⇒
1) local
2) Error Handling

Activation functions ⇒

Hidden layer → Relu ⇒

Output layer ⇒ Sigmoid

Relu → to add Non linearity.

Sigmoid → Linear → X

↓ ↳ log function ⇒ Probability

Choice ↓

↓ Sigmoid ⑦

→

(A) Classification

Log loss.

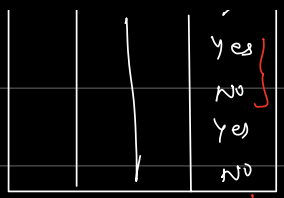
1) Binary Cross Entropy
2) Categorical Cross Entropy
3) Sparse Categorical Cross Entropy.

Categorical Cross Entropy

1)

Target column → One Hot Encoding

①



→ multiple category

↓

[A, B, C, D] → multiple values

10 5

4

②

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

Actual

Predicted value

where k is number of classes in the data

multi

↓

Iteration

Cost = $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}[y_{ij}\log(\hat{y}_{ij})]$

Average

No of iteration

class = 2 = 4

③ One-Hot-Encoding

↳

Yes → 1

No → 0

One Hot Encoding

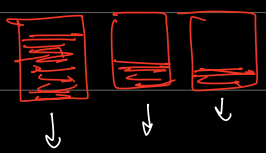
Categorical →

{ (1) → Formula = k = Class
(2) → One Hot Encoding → }

(3) Advantages ⇒ 1) It classify Mutiple class

(4) Disadvantages → (1) One-Hot Encoding for the Target

(2) sensitive Class imbalance



4 = 3

Class imbalance = Accuracy ↓

Activation function : Hidden L → ReLU (Non Linearity)

Output layer → (Softmax) ⊙

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

↓

probability

① Sigmoid → Binary classification

② Multiple class = Softmax AF

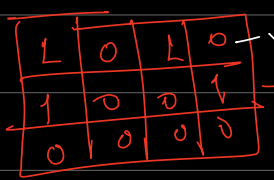
Sparse Categorical Cross Entropy :-

↓

Why sparse ? = To Remove Empty space

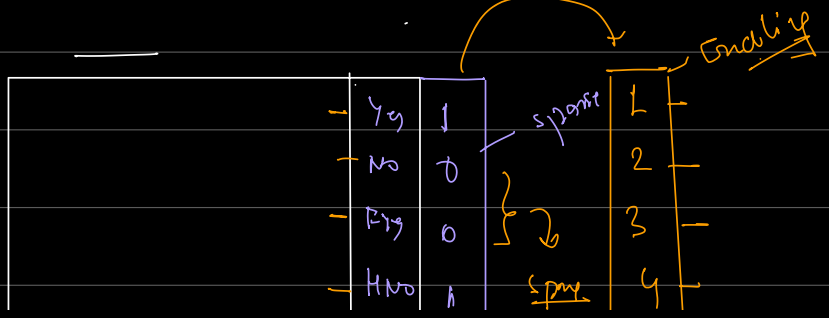
⇒ Purpose → Same for Multi class classification

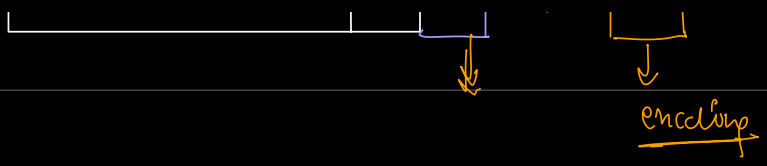
[Sparse Matrix → Empty]



Sparse = value

empty = empty





Basic → Way of doing Encoding

Which → Sparse

1) Formula = Same

Advantage ⇒ (1) → No one Hot Encoding

(2) Computation efficient than Categorical

Disadvantage → 1) It is less intuitive as compare to categorical

(2) Sensitive → Class imbalance

Which to prefer → Intuitive → Categorical

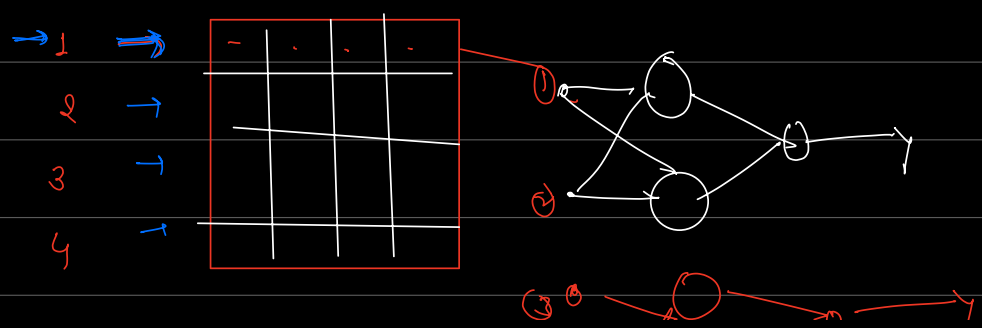
Computation power ⇒ sparse

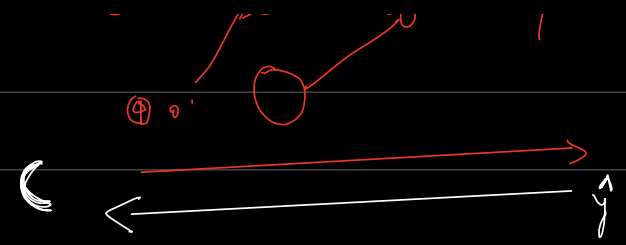
Gradient Descent → Optimization Algorithm

1) Term and Terminalogy
2) Why we Use Gradient
3) Working
4) Types → 1) Batch Gradient Descent
   2) Stochastic Gradient Descent
   3) Mini Batch Gradient Descent
5) Compare → Speed, Convergence, stability
6) When to use which one →.

(1) Terms & Terminology →

1) epoch →





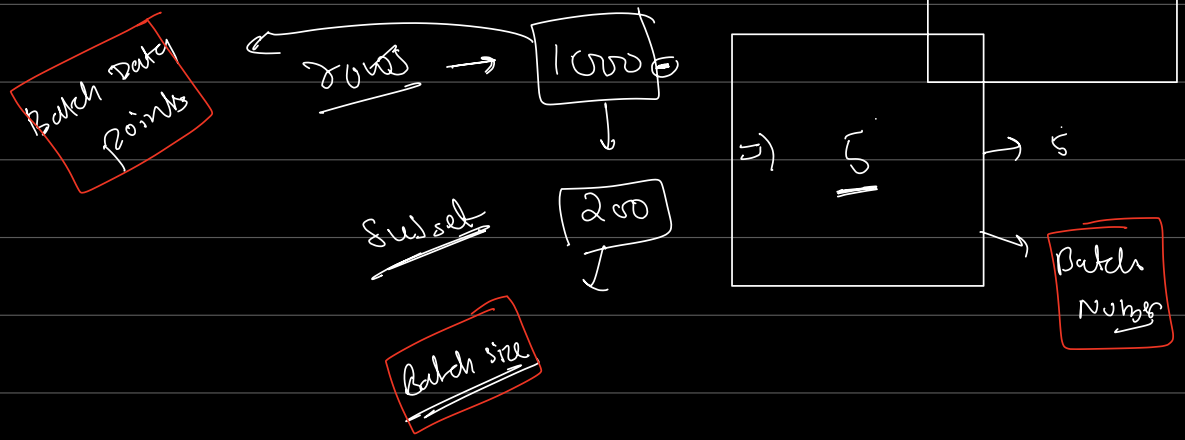
① → For 1 Row ⇒ FP + BP ⇒ iteration

② → Complete data ⇒ FP + BP ⇒ Epoch

③ → Batch ⇒ Collection of Iteration

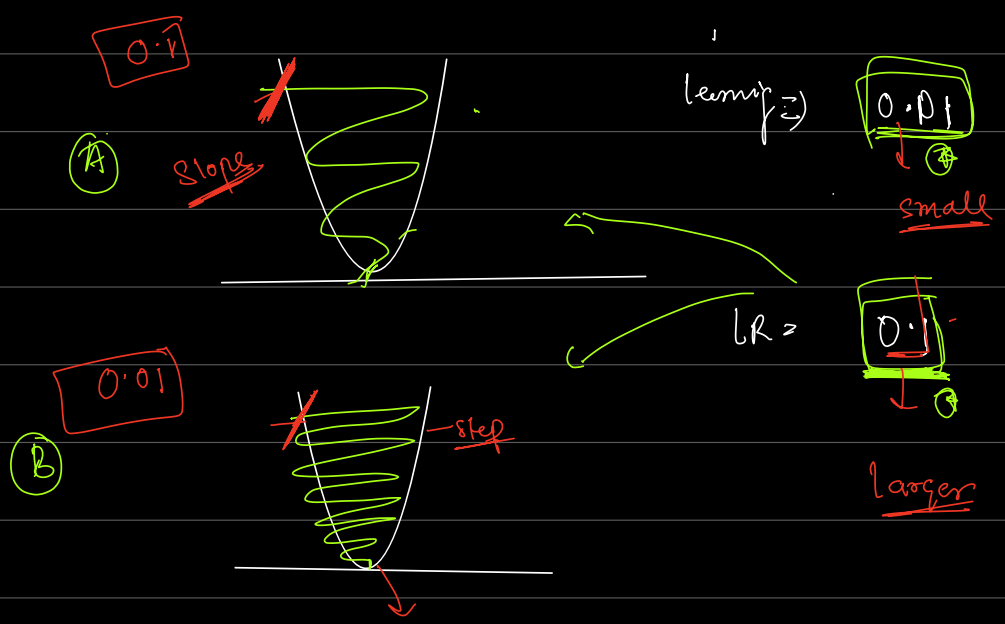
dataset ⇒ 100 data point ⇒ 5

20



④ Learning Rate ⇒ η is step size [optimum the weight & bias]

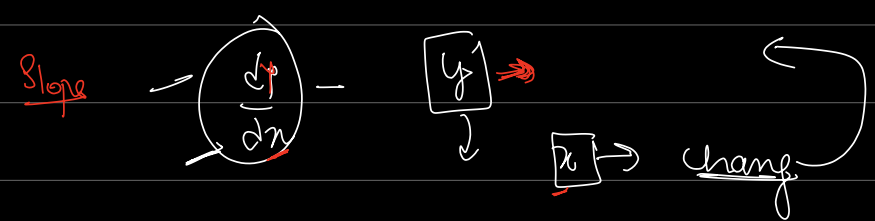
How fast & slow Model learn.



⇒ gradient descent = optimization Algo

Slope ⇒ 1) direction

Slope ⇒ 2) $\frac{dy}{dx} = \frac{y_2 - y_1}{x_2 - x_1}$



$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_i} \times \frac{\partial y_i}{\partial w_1} \Rightarrow \text{Slope}$$

gradient

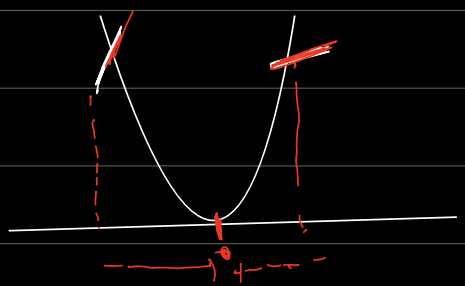
gradient

(2) → Cost function = $\frac{1}{\Sigma}$( )

Gradient Descent → Gradient + Descent

↓ ↓

θ → Downward direction



1)

2) Convergence

↓

Error = 0

Why Gradient Descent :-

Convergence - ✓ 1) It help to convergence :

2) optimization ✓ 2) optimization Algorithm → Try to find optimal value

Scalibity → (3) It can be use both for linear & non-linear models.

efficiency - (4) → It is Computationally efficient.

High dimensional data set

⑤ Efficiency for large dataset.

④ Gradient Desent ⇒ ① optimation Algorithm which is ~~used to~~ minimize the cost function.

loss function ①

Working ⇒

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}}$$

(1) Weight, Bias → Randomly

(2) Loss → Error

(3) Back propagation ⇒ → weight / Bias

To Reduce the Error

Adjust ⇒ optimize

Gradient descent

I/O ← Algo →

⑥ Increase / Decrease the value of W/B

Weight [Bia Updation Formula

Optization ⟹ Error Reduce

1) Gradient Descent — Types
2) —
3) —
4) —
5) —
6) —

Gradient Types → Reason → Batch size

Batch Gradient Descent

epoch - Complete Data

Stochastic Gradient Descent

single datapoint

Mini Batch Gradient D.s.

Collection of Dataset

Batch Gra